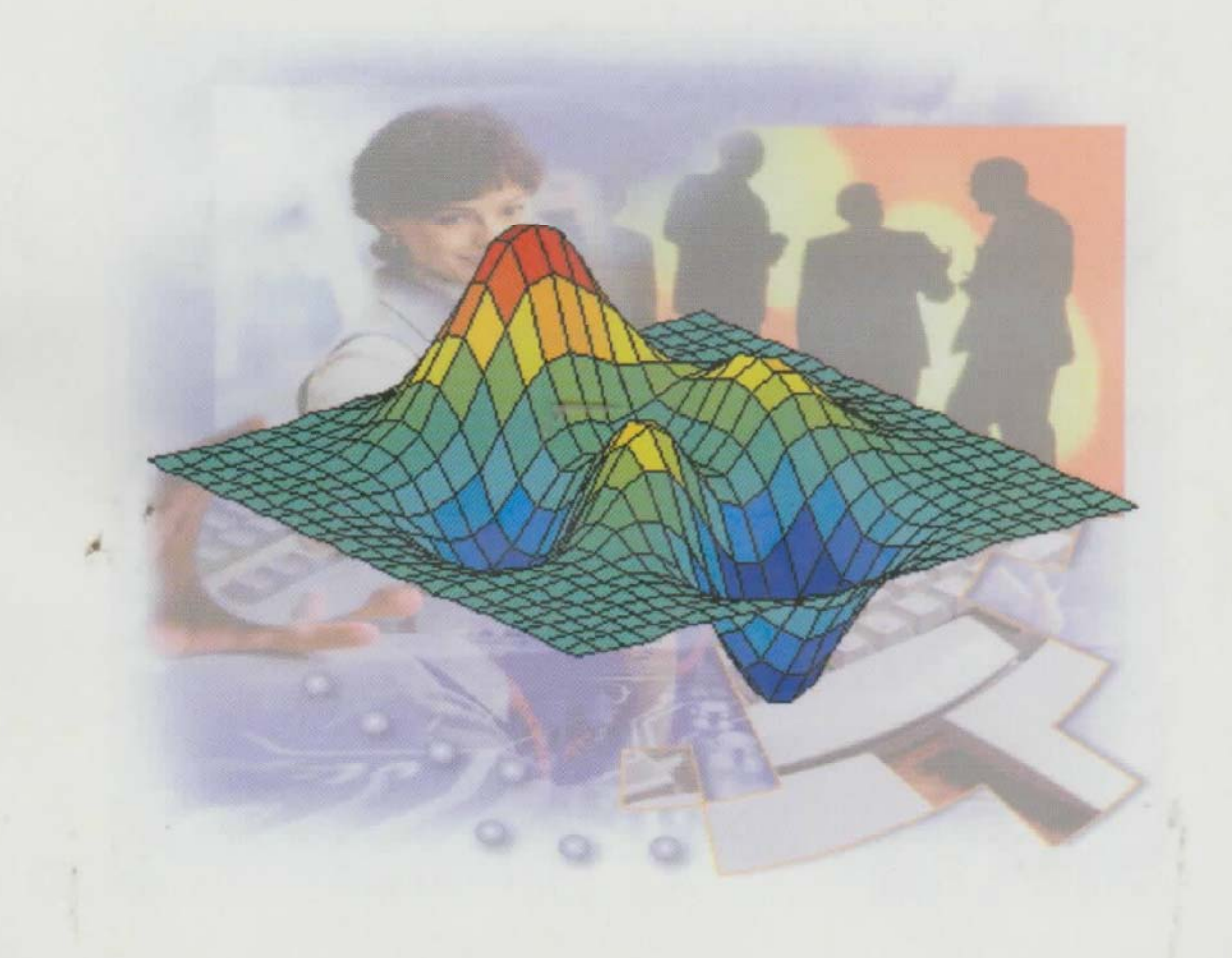
掌握和精通 MATLAB

张志涌　刘瑞桢　杨祖樱　编著



北京航空航天大学出版社

掌握和精通 MATLAB

张志涌　刘瑞桢　杨祖樱　编著



北京航空航天大学出版社

内 容 简 介

本书系统详实地讲述了最新版 MATLAB 的五大通用功能·数值计算功能(Numeric)、符号计算功能(Symbolic)、数据可视化功能(Graphic)、数据图形文字统一处理功能(Notebook)和建模仿真可视化功能(Simulink)。由于这五大功能在命题构思、模型建立、仿真研究、假想验证、数据可视、报告总成和论文撰写各个环节中的非凡能力,使 MATLAB 在线性代数、矩阵分析、数值计算及优化、数理统计和随机信号分析、电路与系统、系统动力学、信号和图像处理、控制理论分析和系统设计、过程控制、建模和仿真、通信系统、财政金融等众多领域的理论研究和工程设计中得到了广泛应用。书中数百个算例将有力地帮助读者理解和掌握 MATLAB 的这五个功能。

本书既可作为理工科院校研究生、本科生系统学习 MATLAB 的教材,又可以作为广大研究人员、工程技术人员掌握和精通 MATLAB 的自学用书。

图书在版编目(CIP)数据

掌握和精通 Matlab/张志涌等编著. - 北京:北京航空航天大学出版社,1997 7

ISBN 7-81012-702-0

Ⅰ 掌… Ⅱ 张· Ⅲ 工程计算程序,Matlab Ⅳ TP317

中国版本图书馆 CIP 数据核字(97)第 04294 号

●书名.掌握和精通 MATLAB
●作者·张志涌 刘瑞桢 杨祖樱
●责编:赵延永
●责校.陈 坤
●出版 北京航空航天大学出版社
●邮编 100083 电话:(010)62015720
●印刷:北京朝阳科普印刷厂
●发行.北京航空航天大学出版社发行
●经售 全国各地书店
●开本:787×1092 1/16
●印张 20 25
●字数:518 4 千字
●印数·4 000 册
●版次 1997 年 8 月第 1 版
●印次 1997 年 8 月第 1 次印刷
●书号·ISBN 7-81012-702-0/TP·243
●定价:28 00 元

前　言

1．MATLAB 及其影响

MATLAB 自 1984 年由美国 MathWorks 公司推向市场以来，历经十几年的发展和竞争，现已成为(IEEE 评述)国际公认的最优秀的科技应用软件。该软件有三大特点：一是功能强大(数值计算和符号计算、计算结果和编程可视化、数学和文字统一处理、离线和在线计算)；二是界面友善、语言自然(以复数矩阵为计算单元，指令表达与标准教科书的数学表达式相近)；三是开放性强(仅 MathWorks 公司本身就推出了 30 多个应用工具箱)。MATLAB 的这些特点使它获得了对应用学科(特别是边缘学科和交叉学科)的极强适应力，并很快成为应用学科计算机辅助分析、设计、仿真、教学乃至科技文字处理不可缺少的基础软件。在国外的高等院校里，MATLAB 已经成为大学生、硕士生、博士生必须掌握的基本技能，在设计研究单位和工业部门，MATLAB 已经成为研究和解决各种具体工程问题的一种标准软件。

MATLAB 广泛流行的另一个表现是，国际上许多新版科技书籍(特别是高校教材)在讲述其专业内容时都把 MATLAB 当作基本工具使用。在我国，应用 MATLAB 的单位和个人也急剧增加。国内一些理工类重点院校已经或正在把 MATLAB 作为攻读学位所必须掌握的一种软件。

2．编写本书的原因

与 MATLAB 在开拓思维空间、激发创造力上的作用相比，与 MATLAB 在世界科技界的地位和影响相比，与 MATLAB 在我国实际流行情况想比，我国关于 MATLAB 的出版物数量和内容都是极不相称的：一是出版物品种极少；二是出版物的内容基本停留在 80 年代中期 MATLAB 3.0 版用户指南的水平上，而对 1993 年以来 MATLAB 迅速发展的一系列重要功能(如符号计算能力、数学文字统一处理能力、可视化建模及仿真能力、动画及图柄操作能力)连提都没有提及。

在国外，除随软件出售的 MATLAB 用户指南外，综合讲述 MATLAB 各大功能的书籍也不多见。如 1995 年初夏国外出版的《MATLAB for Engineers》虽已论及 MATLAB4.0 版中的部分内容，但图柄操作(Handel Graphics)、可视化建模和仿真 Simulink 等重要内容都没提及。据本书作者到 1996 年底为止所搜集到的 90 多本涉及 MATLAB 的国外书籍资料表明，还没有一本书论述过 MATLAB 的符号计算能力(Symbolic)和数学文字统一处理能力(Notebook)。

本书的编写宗旨是：充分、详尽地介绍当今 MATLAB 最新版本(即 4.2 版)的各种通用性功能。这一方面是为了帮助读者用好和用活 1997 年以前的各种 MATLAB 版本；另一方面是为了帮助读者提高适应 MATLAB 升级(5.0 版)和自学新增功能的能力。本书各章节和数百个实例介绍 MATLAB 各种指令的综合运用，较好地弥补了 MATLAB 用户指南和在线帮助的不足。书中适度介绍的 MATLAB 结构和路径管理知识，不仅有助于 MATLAB 软件的安装和

维护，而且有助于对 MATLAB 工作机理的理解。

假如读者借助本书能较快地掌握 MATLAB，并通过 MATLAB 或获得更强认知能力，或从繁杂编程和计算中得到解放，那将使笔者感到极大的欣慰。

3．读者对象

MATLAB 为国际科技界倚重的现实决定了本书的主要服务对象是：研究人员，工程技术人员，理工科高等院校的教师、研究生和本科生。

本书内容不仅仅面向那些对 MATLAB 的使用感兴趣的读者，而且对那些必须参与 MATLAB 安装和维护的读者也将给予切实有效的帮助。

4．内容梗概

本书内容围绕 MATLAB“主包”和它的功能性“工具箱”展开。至于学科性工具箱，不是本书要写的内容。其原因在于．一、MATLAB“主包”和功能性“工具箱”可用于各种学科研究，是学科性工具箱的开发基础；二、学科性工具箱与相应学科的理论知识关系密切，离开专业理论无法解析和讲述学科性工具箱的应用。

全书包括正文和附录两部分。

正文内容

正文包括以下四大块内容：

(1) 国际四大流行科技应用软件简介

这部分内容分布在第一章，第二章的四、五、六节，以及以后几章的“入门”节。

书中在对国际科技计算软件概述基础上着重介绍：MATLAB 的历史和发展现状；分节概略介绍国际上最有影响的另外三个软件：Maple、Mathematica 和 Mathcad。读者即便对这三个软件一无所知，也没有用户指南，也不难循着本书提供的算例和说明，初步领略它们各自的风采，并从中获得一点与 MATLAB 对照的依据。

(2) MATLAB 的结构、安装和配置

这部分内容对希望独自承担 MATLAB 安装、维护和使用全部任务的读者尤为重要。

第二章系统介绍了 MATLAB 的两种不同安装方法：标准安装和非标准安装；还包括如何扩展和修改 MATLAB 的搜索路径以符合用户需要。

除此以外，本书在引入其他功能性工具箱（如符号计算 Symbolic、活笔记本 Notebook、可视化建模和仿真 Simulink）时，还会再次介绍它们与 MATLAB 及其他应用软件（如 Word）的配置要领。

(3) MATLAB 的五大功能

数值计算功能(Numeric)

这是第三章的内容。它包括：矩阵的创建和保存；数值矩阵代数、乘方运算和分解；数组运算；矩阵操作；多项式和有理分式运算；数据统计分析、差分和数值导数；用于求积分、优化和微分方程的数值解的功能函数。

符号计算功能(Symbolıc)

符号计算有两大特点:符号解和任意精度数值解。MATLAB 的符号计算借助符号工具箱(Symbolıc)实现。只有 MATLAB 4 0 及其以后的版本才可能获得这种功能扩展。

第四章用较大篇幅介绍(包括符号矩阵操作、符号微积分、符号代数、代数方程和微分方程的符号解)符号计算 M－函数文件的特点和使用规则,并专门讲述了使用十分方便的“符号函数计算器”。本章还用两节篇幅介绍如何直接利用 Maple 的“核”和“库”进行其他符号计算;如何利用符号计算在线查询系统(包括 MATLAB 符号查询系统和 Maple 符号查询系统)。

图形和可视化功能(Graphıc)

图形和可视化功能是现代应用软件发展的主要方向,也是 MATLAB 4 0 前后版本间的最大差异之一。MATLAB 3 0、3 5 版所具图形功能仅是本章内容的很小一部分。

第五章的大部分篇幅讲述 MATLAB 的(离用户较近的)“高层”图形指令·二维、三维曲线;三维曲面;图形的标识;坐标控制;图形的叠绘;视角和光照设计;色彩精细控制和“第四维”表现力;动态轨迹和影片动画。本章也用一定篇幅介绍了(离用户较远的)“底层”图形指令,以使用图形操纵更得心应手,图形表现更反映事物的本质。

MATLAB 的活笔记本功能(Notebook)

粗略地说,Notebook ＝(MATLAB＋Word)。在 Notebook 环境中,用户不仅拥有 Word 的全部文字处理能力,而且可获得 MATLAB 所赋予的各种数值计算、符号计算和计算结果的可视化能力。

第六章在重点介绍数学指令“细胞”的激活和运作的同时,完整地讲述在 Notebook 环境中文档、数学指令、输出数据和图形的协调操作。

可视化建模和仿真功能(Sımulınk)

自 4 0 版开始,MATLAB 就具备了可视化建模和仿真功能。第七章系统介绍:Sımulınk 方框图模型创建过程中的基本操作;数值分析工具的选择和参数设置;离散系统;一般模型的线性化;平衡点的确定;S－函数运作机理和编制方法;自定义模块的“封装”。

(4) MATLAB 的程序设计和数据的输入输出

MATLAB 除了指令行操作的直接交互使用方式外,MATLAB 作为高级应用软件有它自己的编程语言。要充分体现和发挥 MATLAB 能力,必须掌握 MATLAB 程序的设计。

命令文件和函数文件的基本结构和编制、字符串操作、程序流控制、函数调用和参数传递等都是第八章的内容。

除此以外,第八章的后半部分还将介绍:输入输出数据方法的选择,二进制数据文件和 ASCII 数据文件的存取;不同平台间的文件和数据交换。

附录

本书附录的编制特点是:MATLAB“主包”指令按功能分成若干子类;每个指令都有简明的解释及检索节次。书后附录和书前目录的组合,构成了较完备的检索系统。

5．本书的编写特点

内容新颖全面

据笔者所掌握的(到1997年初为止)资料表明:除用户手册外,以MATLAB 4 2最新版本为基础全面系统介绍MATLAB(除需专门硬件的“实时”功能外)各种重要功能应用的书籍,不仅国内没有,就是在国外也未见到出版消息。

一般归纳和算例并重

本书在对功能、指令作一般描述的同时,提供几百个算例。书中所有算例的程序、指令和算得的文字结果及图形,都是笔者运作的真实记录,给读者以切实、可重复的参照样板,减少读者对新知识新指令的不确定感。

内容层次丰富,适应面宽

本书内容分三个层次:入门、基本、深入。从总体上讲,不同层次的内容均按由易到难的原则编排。内容涵盖面从四大国际流行科技软件入门到高水平的图柄操作,从数学符号计算到MATLAB及其工具箱的环境配置和维护。

系统论述和快速查阅兼顾

本书所有章节构成对MATLAB各大功能的系统全面讲述,但就每章每节内容而言,它们又相对独立。因此,本书既可系统学习,又可随时查阅。详细目录和书后附录为读者提供了方便的查阅手段。

MATLAB的Notebook是本书稿的支柱软件

Word是本书成稿的最基础环境。除Maple、Mathematica、Mathcad、动画等极少数内容外,本书的其他内容都在Notebook(MATLAB＋Word)的集成环境下直接写成。因此,假如读者也有同样的工作环境,那么书中的指令都是“活”的,即本书内容都可以在用户的环境中重现。

6．使用本书的知识准备

无论是对那些仅仅在MATLAB环境中工作的读者,还是对独自承担MATLAB安装、使用和维护各环节工作的读者,本书内容都是相对自完备的,即不需要读者在使用本书之前有什么特别的计算机知识准备。即便是对那些希望在数学文字综合环境中工作的读者,本书也配备了最简明的Word 6.0使用引导,以确保他们顺利地运用MATLAB的“活笔记本”Notebook。

MATLAB是数学软件,因此本书的读者应事先懂得有关的数学定义、术语、规则及用途。但这并不意味着,用户要完全了解实现那些数学运算的细节。

7．致谢

笔者自1989年接触MATLAB开始,于1992年受福州大学科研基金专项资助,后于1993年、1995年又先后两次受到福建省教委科研基金和福建省自然科学基金资助。在这些基金资助下,笔者搜集了比较全面的MATLAB及其他科技软件资料,完成了对MATLAB3 0版的汉

化和内部讲义的编写，实现了对 MATLAB 发展的跟踪、消化及对照研究。尽管笔者学习和应用 MATLAB 已有几年，也尽管本书叙述谨慎、算例几经反复运作，但由于本书的结构、内容和算例都融入了笔者的一孔之见，恐难全显集数学、软件和其他学科知识于一体之 MATLAB 的真谛。望使用 MATLAB 的专家、同仁和广大用户不吝赐教。

无论是在 MATLAB 的研究中，还是在本书的成稿过程中，福州大学自动化所和福州大学图书馆网络中心的余泽胜、张顺利、詹庆东等软件研究人员都给予笔者持久而有力的支持。

此外，在本书成稿过程中，笔者受到国内许多学者、专家和研究人员的帮助和鼓励。在正式出版过程中，又受到北京航空航天大学出版社的鼎力相助。

在本书出版之际，笔者向福州大学、福建省教委和科委、北京航空航天大学出版社，向给予我们支持和鼓励的所有学者、老师和同仁表示最诚挚的感谢和深深的敬意。

作　者

1997 年元月于福州大学

目 录



第一章　概　论

在当今30多个数学类(为区别于文字处理和作图类而加的修饰词)科技应用软件中,就软件数学处理的原始内核而言,可分为两大类。一类是数值计算(Number Crunching)型软件,如MATLAB、Xmath、Gauss、MLAB等。这类软件对大批数据具有较强的管理、计算和可视化能力,运行效率高。另一类是数学分析(Math Analysis)型软件,如Mathematica、Maple、Macsyma等。它们以符号计算见长,并可得到解析符号解和任意精度解,但处理大量数据时运行效率较低。经过多年的国际竞争,MATLAB已经占据了数值型软件市场的主导地位,处于其后的是Xmath;而Maple、Mathematica、Macsyma位居符号软件的前三名(见1995 IEEE Spectrum)。

在国际流行的科技应用软件中,Mathcad别具特色。该软件的开发商Mathsoft公司一开始就把面向教学和办公作为Mathcad的市场目标。在对待数值计算、符号分析、文字处理、图形能力的开发上,不以专业水准为追求,而尽力集各种功能于一体。

MathWorks公司顺应多功能需求之潮流,在其卓越数值计算和图视能力的基础上,又率先在专业水平上开拓其符号计算、文字处理、可视化建模仿真和实时控制能力,精心营造适合多学科、多部门要求的新一代科技应用软件MATLAB。

本章共分四节。第一节简述MATLAB的发展历史、内容和影响。后三节分别简介Maple、Mathematica和Mathcad的概况及入门材料。

1.1　MATLAB简介

1.1.1　MATLAB是什么

MATLAB原先作为矩阵实验室(Matrix Laboratory),是用来提供通往LINPACK和EISPACK矩阵软件包接口的。后来,它渐渐发展成了通用科技计算、图视交互系统和程序语言。

MATLAB的基本数据单位是矩阵。它的指令表达与数学、工程中常用的习惯形式十分相似。比如,矩阵方程b=Ax,在MATLAB中被写成b=A*x。而若要通过A,b求x,那么只要写x=A\b即可,完全不需要对矩阵的乘法和求逆进行编程。因此,用MATLAB解算问题要比用C、FORTRAN等语言简捷得多。

MATLAB已经受了用户的多年考验。在欧美高等院校,MATLAB已经成为应用线性代数、自动控制理论、数理统计、数字信号处理、时间序列分析、动态系统仿真等高级课程的基本教学工具;成为攻读学位的大学生、硕士生、博士生必须掌握的基本技能。在设计研究单位和工业部门,MATLAB被广泛地用于研究和解决各种具体工程问题。

现在,MATLAB已经成为一个系列产品:MATLAB"主包"和各种可选Toolbox"工具包"。主包中有数百个核心内部函数。迄今所有的三十几个工具包又可分为两类:功能性工具包和

学科性工具包。功能性工具包主要用来扩充 MATLAB 的符号计算功能、图视建模仿真功能、文字处理功能以及与硬件实时交互功能。这种功能性工具包能用于多种学科。而学科性工具包是专业性比较强的，如控制工具包(Control Toolbox)、信号处理工具包(Signal Processing Toolbox)、通信工具包(Communication Toolbox)等都属此类。

开放性也许是 MATLAB 最重要、最受人欢迎的特点。除内部函数外，所有 MATLAB 主包文件和各工具包文件都是可读可改的源文件，用户可通过对源文件的修改或加入自己编写文件去构成新的专用工具包。

MATLAB 有专业版和学生版之分。这两个版本功能上差别不大。学生版只能计算规模有限、难度有限的问题，但价格要比专业版低得多。专业版由 MathWorks 公司发行，学生版则由 Prentice－Hall 出版公司发行。

1.1.2 MATLAB 的发展历史

MATLAB 诞生和 MathWorks 公司的成立

在 70 年代中期，Cleve Moler 和其同事在美国国家科学基金的资助下研究开发了调用 LINPACK 和 EISPACK 的 FORTRAN 子程序库。LINPACK 是解线性方程的 FORTRAN 程序库，EISPACK 则是解特征值问题的程序库。这两个程序库代表着矩阵计算软件的最高水平。

到 70 年代后期，身为新墨西哥大学计算机科学系主任的 Cleve Moler，在他给学生开线性代数课程时，想教学生使用 LINPACK 和 EISPACK 程序库，但又不希望学生在 FORTRAN 编程上花太多时间，因为这不是他开课的目的。于是，他开始用业余时间为学生编写方便使用 LINPACK 和 EISPACK 的接口程序。Cleve Moler 给这个接口程序取名为 MATLAB，这是从 MATrix，LABoratory 各取前三个字母组成的，意思是“矩阵实验室”。

又过了几年，有一次 Cleve Moler 应邀去另一所大学讲学。在访问结束时，他把 MATLAB 留在了那所大学的计算机上。从那以后，经一二年时间，MATLAB 开始受到欢迎，并成了应用数学界的术语。

1983 年早春，由于 Cleve Moler 对斯坦福大学的访问，Jonh Little 受到了 MATLAB 的影响。作为工程师的 Little 觉察到 MATLAB 的潜在应用天地是工程领域。同年，他与 Moler、Steve Bangert 一起合作开发第二代专业版 MATLAB。从这一代开始，MATLAB 的核心就采用 C 语言编写。也是从这一代开始，MATLAB 不仅具有数值计算能力，而且具有了数据图视功能。

1984 年成立的 MathWorks 公司正式把 MATLAB 推向市场，并继续从事 MATLAB 的研究和开发。

现在(指 1997 年初本书完稿时)，Jonh Little 是 MathWorks 公司总裁，Cleve Moler 是 MathWorks 公司首席科学家。

MATLAB 的发展

在 MATLAB 进入市场前，国际上的许多应用软件包都是直接以 FORTRAN、C 等编程语

言直接开发的。这种软件的缺点是适用面窄、接口简陋、程序结构不开放以及没有标准的基库，很难适应各学科的最新发展，因而也很难推广。

MATLAB 在市场上的出现，为各国应用科学家开发学科软件提供了新的基础。例如在 MATLAB 问世不久的 80 年代中期，原先控制领域里的一些封闭式软件包（如英国的 UMIST、瑞典的 LUND 和 SIMNON、德国的 KEDDC）就纷纷淘汰或在 MATLAB 上重建。

MATLAB 的应用又反过来推动其发展。表1 1-1 是近年来 MATLAB 的版本更新简况：

表 1.1-1 MATLAB 版本升级历程

日 期	版 本	平 台	MATLAB 系列的重要工具包软件
1987 年	MATLAB 3 0 版	Dos	Control, Signal, Identification
1991 年	3 5 版	Dos	图形编程、仿真软件 Simulib（Simulink 的前身），Optimization, Robust Control, Neural Networks
1993 年	3 5k 版	Windows 3 0	
1993 年 1 月	4 0 版	Windows 3 x	Matlab with Simulink, Control, Neural Networks, Optimization, Robust Control, Signal Processing, Spline, State-space Identification, System identification, μ-analysis and synthesis
1993 年 11 月	4 1 版		Symbolic Math 符号计算工具包
1994 年 5 月	4 2 版		DSP Blockset
1994 年 11 月			Notebook for Word "活"笔记本工具包，Real-Time Workshop
1995 年 5 月	4 2c 版	Windows 3 x	Fixed-Point Blockset
1996 年 4 月			MATLAB Compiler, C Math Library
1997 年夏	（预计）Matlab 5 0; Simulink 2 0	Windows 95	（MathWorks 公司预计 ）在继承 Matlab 4 2c 和 Simulink 1 3c 版本功能的基础上，实现真正的 32-bit 运作。数值计算更快、图形表现更有效、编程更简捷直观、用户界面更友善。

注 本表中的日期、版本仅就 PC 机而言，对其他机种而言，表中内容会有出入。

MATLAB 适配的操作系统平台

据 1996 年 11 月的资料表明，MATLAB 4 2c 现适配于以下平台：

IBM-PC	Windows 3 x, Windows NT 3 5, Windows 95, OS/2 2 1
Macintosh	68k 和 PowerMac
SUN	Sun OS 4 1.3 和 Solaris 2 4
HP9000/700	HP-UX 9 05
HP9000/300	HP-UX 9.03
SGI R4000/8000	Irix 5 2 和 6 0
IBM RS/6000	AIX 3 2 5
DEC R3000	Ultrix 4 4
DEC Alpha AXP	Digital UNIX 3 0
PC	Linux 1 2

DEC Alpha AXP　　VMS 6 1
DEC VAX　　VMS 6 1

1.1.3 MATLAB 系列产品及应用

MathWorks 公司的 MATLAB 系列产品

现在世界上以 MATLAB 为基础开发的专业性应用软件和硬件(包括一些实验装置)很多。在表 1 1－2 中所列的产品仅是 MathWorks 公司自已推出的软硬件(包括 MATLAB、Toolboxes、Blocksets 三大类)。该表据 1996 年底资料编制而成。

表 1.1-2 MathWorks 公司的 MATLAB 系列产品

产品名称	控制系统	信号处理	数据分析	通信系统	金融系统	工程数学	图形可视
MATLAB	P	P	P	P	P	P	P
Notebook for Word	P	P	P	P	P	P	P
MATLAB Compiler	P	P	S	P	S	P	
MATLAB C Math Library	P	P	S	P	S	P	
Simulink	P	P	P	P	P		P
Symbolic Math	S	S	P	S	S	P	P
Chemometrics			P				P
Communication	S	S		P			
Control System	P				S		
Financial			S		P		S
Frequency Domain System Identification	P	P	P				P
Fuzzy Logic	P	P	P		S		P
Higher-Order Spectral Analysis		P	P				
Image Processing		P	P				P
LMI Control	P		S			S	
Model Predictive Control	P						
MMLE3 Identification	P						
μ-Analysis and Synthesis	P						
NAG Foundation			P		P	P	
Neural Network	P	P	P	P	P		P
Optimization	P	P	P	P	P	P	P
Partial Differential Equation			P			P	P
QFT Control Desigh	P						
Robust Control	P						
Signal Processing	P	P	P	S	S		P
Spline	S	S	P		P	S	P

表 1.1-2(续)

产品名称	控制系统	信号处理	数据分析	通信系统	金融系统	工程数学	图形可视
Statistics	S	S	P	S	P	P	P
System Identification	P	S	P	S	S	S	P
Wavelet	S	P	P	S		S	P
DSP Blockset	P	P					
Fixed-Point Blockset	P						
Nonlinear Control Desigh Blockset	P						
Real-Time Workshop	P	P			S		
RTW Ada Extention	P	P			S		
Simulink Accelerator	P	P			S		

注 P 表示主用学科，S 表示辅用学科。

有关 MATLAB 的书籍

从 80 年代末起，把 MATLAB 作为介绍和教学内容的专业书籍开始陆续出现。到 1996 年底，据 MathWorks 公司自己宣布，在世界上包含 MATLAB 内容或把 MATLAB 作为基本工具的高等专业书籍已达 150 种。在此罗列若干，供读者参考。

《Digital Filters and Signal Processing》
Leland B Jackson
Kluwer Academic Publishers, 1989

《Analog and Digital Control System Design Transfer-Function, State-Space, and Algebraic Methods》
Chi-Tsong Chen
Oxford University Press, 1993

《Applied Factor Analysis in the Natural Sciences》
Richard A Reyment and K G Joreskog
Cambridge University Press, 1993

《Engineering Problem Solving with MATLAB》
D M Etter
Prentice Hall, 1993

《Engineering Vibration》
Daniel J. Inman
Prentice Hall, 1994.

《Modeling of Dynamic Systems》
Lennart Ljung and Torkel Glad

Prentice Hall, 1994

《Numerical Methods for Physics》
Alejandro Garcia
Prentice Hall, 1994

《Probability, Random Processes, and Estimation Thoery for Engineers》
Henry Stark and John W Woods
Prentice Hall, 1994

《Robust Industrial Control》
M J Grimble
Prentice Hall, 1994

《Control Systems Engineering》
Norman S Nise
Addison-Wesley, 1995

《MATLAB for Engineers》
Adrian Biran and Moshe M G Breiner
Addison-Wesley, 1995

《Modern Communication Systems Principles and Applications》
Leon W Couch II
Prentice Hall, 1995

《Neural Network Design》
Martin T Hagan, Howard B Demuth, Mark Beale
PWS Publishing Company, 1996

《Statistics Digital Signal Processing and Modeling》
Monson Hayes
John Wiley and Sons, Inc , 1996

《Wavelets and Filter Banks》
Gilbert Strang and Truong Nguyen
Wellesley-Cambridge Press, 1996

1.2 Maple V 的概述

详细介绍 Maple 不是本节的目的，因此本节将不对 Maple 的工作环境、特点、指令和语言多加解释，只是列举若干可实际运作的典型算例。假如读者有 Maple 软件，那么本节的内容也许能帮助您进入 Maple 软件之门。关于符号运算更详细的论述，可以参阅第五章和 Maple 的用户手册。

1.2.1 Maple 是什么软件

Maple V 是由加拿大 Waterloo University 发展起来的一种数学软件，它那无与伦比的符号运算能力，已使 Maple 在国际通用数学软件的激烈竞争中独占鳌头。不仅是国外学生中广为流行的“科学便笺式”软件 Mathcad 靠 Maple 实现符号运算，就连首屈一指的数值计算软件 MATLAB 在扩展其符号运算功能时，也借助了 Maple 的威力。

Maple V 版本提供数学函数 2000 余种，其涉及范围：基本代数学、欧几里得几何学、数论、有理函数、微积分、线性代数及矩阵论、微分方程、图形学、离散数学、群论以及数学的其他许多领域。此外，在数值计算和图形处理上，Maple 也有相当强的实力。Maple 是一个开放的系统，它提供一套内部的编程语言，使用户可以开发自己的应用程序。事实上 Maple 所提供的 2000 多种函数中，绝大部分是用这种语言编写的。

1.2.2 Maple 的工作窗口

Maple V Release 2 有基于 DOS 和 Windows 的两个版本，可以在 80386 或 80486 以上的微机上运行。基本运行环境要求硬盘有至少 15MB 空间、4MB 的内存，其中 Windows 版本需要 MS-Windows 3.1 以上版本(中西文版本均可)。

Maple V for Windows 启动后会出现一个 Maple 工作区空白窗口，如图 1.2-1 所示。

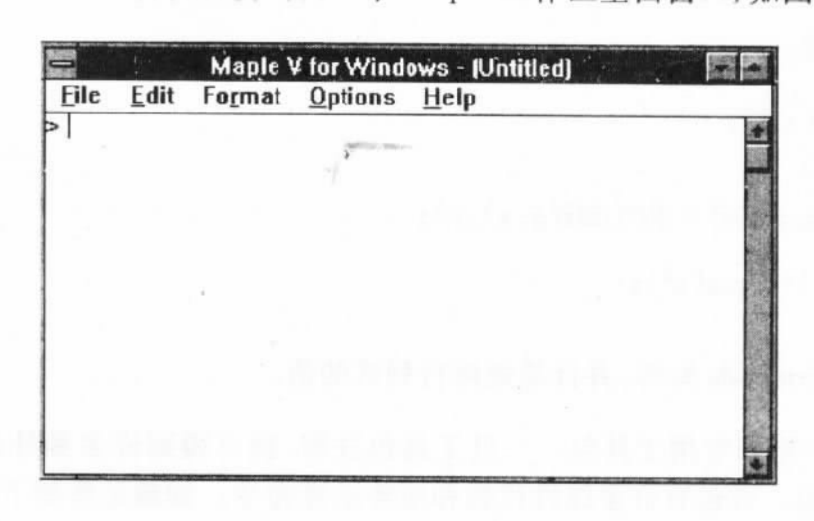


图 1.2-1 Maple 工作窗口

此时，用户就可以在 Maple 工作区内出现的提示符号“>”后，输入命令，进行工作。需要说明的是：输入表达式后面必须用分号“;”结尾，以表示命令的结束，然后按回车键进行运算。如若结尾缺少“;”，那么 Maple 将认为命令没有结束，而处于等待状态。希望用户切记 Maple 的这一特征。

1.2.3 符号运算

Maple 最主要的功能是符号运算，其功能强大是举世公认的。符号运算的魅力在于：运算

时，无须事先对独立变量赋值，而所得的结果以标准的符号形式表达。在符号表达式中的数字都是不含任何机器精度误差的绝对准确值(如：2/3 就绝不会表示成 0.66666…)。这是任何数值计算所无法实现的。

【例 1】有理分式运算(包括展开和化简)。

```
>f:=expand((41*x^2+x+1)^2*(2*x-1))/expand((3*x+5)*(2*x-1));
```

$$f:=\frac{3362x^5-1517x^4+84x^3-79x^2-1}{6x^2+7x-5}$$

```
>normal(f);
```

$$\frac{1681x^4+82x^3+83x^2+2x+1}{3x+5}$$

【例 2】解单个代数方程(此时表达式总是假设等于零)。

```
>x^3+1/2*x^2*a+13/3*x^2-13/6*x*a-10/3*x+5/3*a;
```

$$x^3-\frac{1}{2}x^2a+\frac{13}{3}x^2-\frac{13}{6}xa-\frac{10}{3}x+\frac{5}{3}a$$

```
>solve(",x);
```

$$-5,\frac{2}{3},\frac{1}{2}a$$

说明：第二个指令圆括号内的双引号代表此指令前的最新运算结果。

【例 3】求高阶导数。

```
>g:=y*sin(x^2);
```

$$g:=y\sin(x^2)$$

```
>Diff(Diff(g,x),y)=diff(diff(g,x),y);
```

$$\frac{\partial^2}{\partial y\partial x}y\sin(x^2)=2\cos(x^2)x$$

【例 4】给出 Vandermonde 矩阵，并计算矩阵行列式的值。

Maple 带有一系列专用工具包。一旦工具包驻留，就可得到许多额外命令。如可用 with()命令线性代数包。该包有许多线性代数和矩阵运算命令。加载运作如下：

```
>with(linalg);
>V:=vandermonde([u,v,w]);
```

$$V:=\begin{bmatrix}1 & u & u^2\\ 1 & v & v^2\\ 1 & w & w^2\end{bmatrix}$$

```
>d:=det(v);
```

$$d:=vw^2-v^2w-uw^2+u^2w+uv^2-u^2v$$

1.2.4 数值计算

求完全准确的符号解的代价是:对所解问题的限制、所给解的不彻底性、运算时间开销大。为解决这些问题, Maple 也提供自己的数值计算能力。

对于那些有最终符号解的问题, Maple 可根据用户需要把符号结果变换成任何精度的数值结果。对那些只有中间符号解结果的问题,用户可以用数值运算来获取终极解。这样做的另一个好处是:尽量缩短机器截断误差的传递路径。

【例 1】准确值运算及浮点值近似解的求取。

```
>(2^30/3^20)*sqrt(2);
```

$$\frac{1073741824}{3486784401}\sqrt{2}$$

```
>evalf(");
.4355016184
```

【例 2】有限(或无限)求和运算及数值解。

```
>Sum((1+i)/(1+i^4),i=1..20);
```

$$\sum_{i=1}^{20}\frac{1+i}{1+i^4}$$

```
>value(");
27531089155139512664184734944799665266675157706349981434584200143/
    21653762264974190884124027285387789492160976952723410943046016049
```

1.2.5 图形功能

Maple 有二维及三维图形支持计算结果的可视化。

【例 1】plot3d()提供三维图形(函数可以是表达式,程序,语句函数或点列)。

```
>plot3d(x*exp(-x^2-y^2),x=-2..2,y=-2..2,orientation=[80,57]);
```

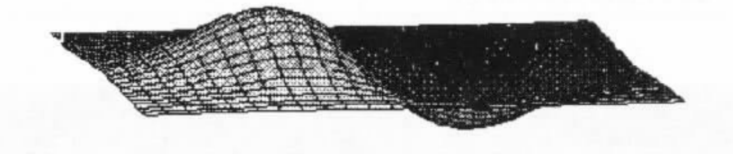


图 1.2-2 Maple 所画三维曲面

1.3 Mathmatica 概述

本节简单介绍另一个著名软件 Mathmatica 。像上节一样，以实际可操作的算例进行介绍。读者借助算例，可以初涉 Mathmatica 。

1.3.1 什么是 Mathmatica

Mathmatica 的原始系统是由美国物理学家 Stephen Wolfram 领导的一个小组开发来进行量子力学研究的。软件开发的成功促使 Stephen Wolfram 于 1987 年组建 Wolfram 研究公司，并推出了该公司的商品软件 Mathmatica 1.0 版。此后，Wolfram 公司通过对 Mathmatica 的不断改进和扩充，陆续推出了 1.2 版、2.0 版和 1996 年的 3.0 版。目前，国内最常见的是 2.0 版。

Mathmatica 拥有范围广泛的数学计算功能，支持比较复杂的符号计算和数值计算。因此，早期它主要在数学、物理等理论研究领域中流传。近几年，Wolfram 公司为帮助工程技术人员克服直接运用 Mathmatica 时所遇的困难，正在开发以 Mathmatica 为基础的各种应用软件包，已经推出的有电气工程软件包、小波分析软件包等。

Mathmatica 的基本系统是 C 语言编写的，因此能方便地移植到各种计算机系统上。目前常见的是：MS-DOS 386 版、MS-DOS 386/387 版和 MS-Windows 版本。后者利用 Windows 环境，使用方式和用户界面都有了重大改进，如图 1.3-1 所示。

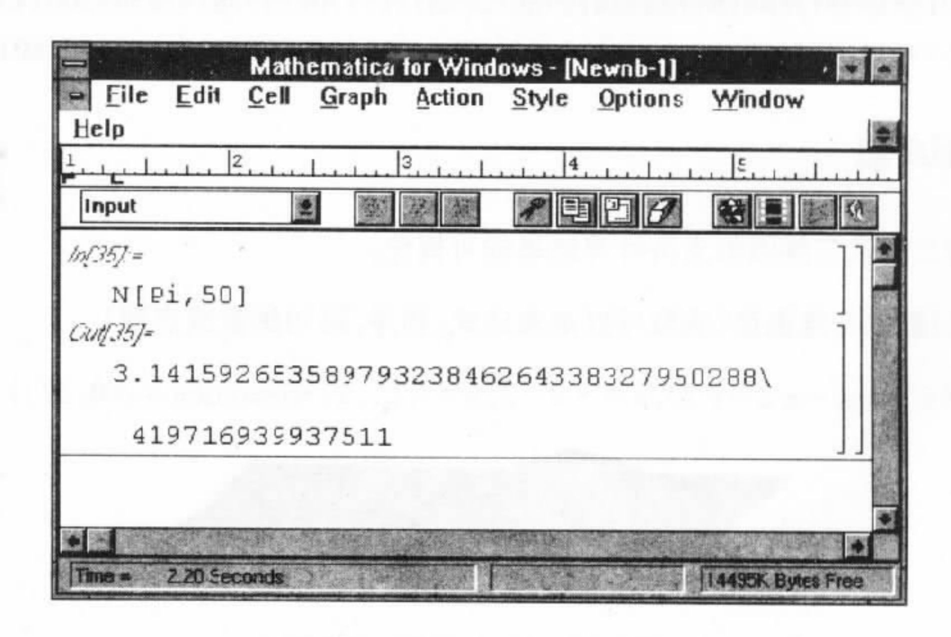


图 1.3-1 Mathmatica 的工作区窗口

说明：Mathmatica 将输入的指令用标题“In[n] = ”标识，输出结果用“Out[n]”标识，其中数字“n”表示已经输入的指令数。

1.3.2 数值计算

Mathmatica 可以对整数及有理数进行准确运算。

【例 1】乘方运算。

(输入以下指令)

In[1]:=

```
22^34
```

(系统将回答)

Out[1]=

```
4389056261830591470007906571986683114651910144
```

说明：Mathmatica 的 Windows 版是将表达式作为一个单元体，运行输入的表达式是同时按住【Shift + Enter】键，而不是通常的只按【Enter】键。为简单起见，Mathmatica 窗口显示的标题“In[n]:=”和“Out[n]=”在例子中直接写出来。

【例 2】开方运算。

In[2]:=

```
%^(1/34)
```

Out[2]=

```
22
```

说明：Mathmatica 用符号“%”表示最近一次的计算结果，“%%”表示倒数第二次的结果，依次类推。另外一种通用的表示法是用符号“%”后面紧跟一个整数表示以该整数为序号的那次计算结果。

1.3.3 符号计算

Mathmatica 的一个重要特点是能对符号表达式进行处理，下面是一些符号计算的例子，可以从这些例子中体会 Mathmatica 符号计算的能力。

【例 1】高阶导数运算。

In[3]:=

```
D[Sin[x]/(x+Cos[[2*x]), {x, 3}]
```

Out[3]=

$$\frac{6\mathrm{Cos}[x]}{x^3}-\frac{\mathrm{Cos}[x]}{x}-\frac{6\mathrm{Sin}[x]}{x^4}+\frac{3\mathrm{Sin}[x]}{x^2}+8\mathrm{Sin}[2x]$$

说明：这里“D”是求导操作。在求导的表达式中，如有多于一个的变量，则进行的是求偏导数操作。

【例 2】不定积分。

```
In[4]:=
        Integrate[(x+3)^2+2*x+3, x]
Out[4]=
```

$$12x+4x^2+\frac{x^3}{3}$$

【例 3】解代数方程。

```
In[5]:=
        Solve[x^3-1==0, x]
Out[5]=
```

$$\{\{x->1\},\{x->(-1)^{2/3}\},\{x->(-1)^{4/3}\}\}$$

1.3.4 图　形

Mathmatica 的图形功能相当强大，支持各种函数图形，包括二维和三维一般显函数图形和参数形式图形。Mathmatica 系统还有一个与众不同的图形操作特点：图形可以像构造其他表达式一样构造，也可以像定义其他函数一样被定义。

【例 1】三维图形(图 1.3-2)。

```
In[6]:=
        Plot3D[x^2*Sin[y], {x,-1,1},{y,0,Pi}]
```

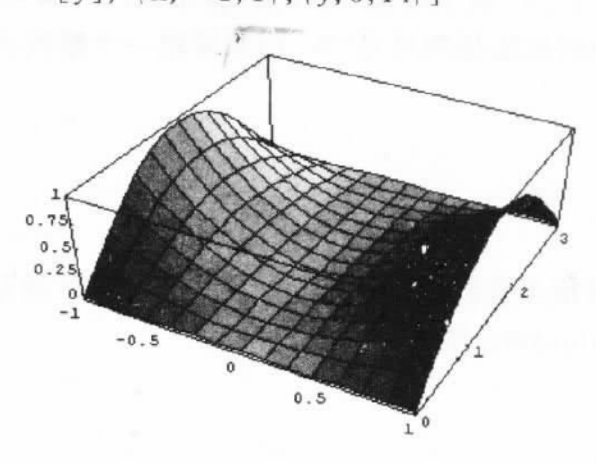


图 1.3-2　Mathmatica 所画的三维曲面

1.4 Mathcad 概述

本书要简单介绍的第三个数学软件是在国外学校、机关、银行、公司中较为流行的数学软件 Mathcad 。该软件的风格及服务宗旨与前两个软件有较大的不同。

1.4.1 什么是 Mathcad

Mathcad 是 80 年代出现得较早的一个交互式数学文字软件。其开发商是 MathSoft 公司。该软件的市场定位是:向广大教师、学生、工程人员提供一个兼备文字、数学和图形处理能力的集成工作环境,以使他们能方便地准备(自然科学)教案、完成(自然科学)作业和准备科学分析报告。需讲究精度、速度、算法稳定性的复杂数值计算问题和需经复杂推理的符号计算问题,都不是 Mathcad 所致力解决的目标。

自问世以后,MathSoft 公司对它进行不断的完善和扩充,新版本一个接一个地推向市场,直到 1996 年的 Mathcad 7.0 版。现今,新版 Mathcad 的计算能力已远远超出了该软件早期的设计目标。

Mathcad 有三大"面向大众"的特点。(1)人们按习惯的标准书写格式输入数学公式、方程组和矩阵后,计算机就能直接给出(或数字、或符号、或图形)结果。用户无须考虑方法以及中间步骤,整个过程就像使用计算器一样简单。(2)灵活的"便笺"式文字处理能力。(3)由 Mathcad 生成的"(Electric Book)电子书籍"中的指令、函数、图形都是"活"的,即指令中任何参数的变化都会使相应结果改变。

1.4.2 Mathcad 的工作窗口

到目前为止,Mathcad 已经发展了很多版本,并且由于商业的需要,分为学生版、标准版、增强版等不同版本。它们之间的功能略有差别,本节将对 Mathcad 6.0 增强版(Mathcad 6.0 Plus)的功能做一个简单的介绍。

Mathcad 6.0 启动后会出现一个 Mathcad 工作区窗口,窗口上方的操作板有许多标准的数学符号供用户选择,以方便地写出所需的数学表达式,见图 1.4-1。首先在工作区内用鼠标点击(click)一下,工作区内便出现一个红十字,它提示:用户可以开始工作。

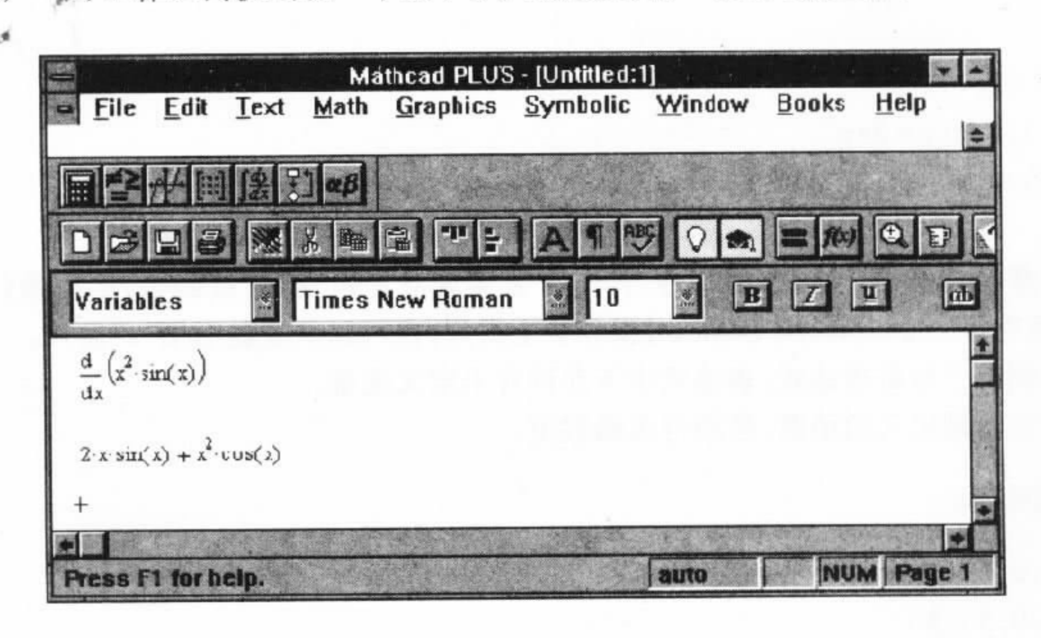


图 1.4-1 Mathcad Plus 6.0 工作区窗口

1.4.3 Mathcad的工作特点

作为概要性介绍，本节不介绍Mathcad的文字处理，只介绍它的数学工作特点，并且以实例进行介绍。

【例1】基本数值运算及内部函数。

$$\sqrt{\frac{1\ \ 837\cdot 10^{3}}{100+3^{5}}}=2\ \ 3142353232$$

说明：

(1)等式左边内容的生成过程：从操作板上得到平方根号，然后依次键入数字和算符。“加”、“减”、“乘”、“分式号”分别用键盘(或工具模板)上的算符“+”、“-”、“*”和“/”送入，并且在屏幕上显示成上述(等号左边)习惯的形式。

(2)在左边输完后，键入“=”，便可得到结果。“=”有两重含义：在该式输入过程中，“=”的键入就执行对其左边表达式指令的运算；在运算结束后，“=”就体现“等于”的含义。

(3)本例中的根号是Mathcad的内部函数。Mathcad有许多内部函数，它们可从菜单条上【Math】栏的下拉菜单中通过【Choose Function】选择获得，也可通过键盘和工具模板上提供的各种数学符号写成。

【例2】物理单位处理能力。

$$\frac{2350\cdot \mathrm{km}}{1\cdot \mathrm{hr}}=652\ \ 78\cdot \mathrm{m}\cdot \mathrm{sec}^{-1}$$

说明·假如输入部分有物理单位，Mathcad会自动进行单位换算，并在结果中给出。

【例3】Mathcad允许用户自定义变量、表达式和函数。

$a:=4$

$a+\sqrt{a}=6$

$g(x,y):=x\cdot y+3\cdot y$

$g(2,3)=15$

说明：

(1)注意本例第一、三行中的符号“:=”。它实现对左边变量(或函数)的赋值(或定义)。该符号既可在【Evaluation and Boolean】操作板上得到，也可以从键盘上用“:”输入。

(2)本例第二行是表达式，表达式中不允许有未定义变量。

(3)第三行是定义的函数，第四行求函数值。

【例4】序列计算。

$x:=0,0\ 5\ \ 2$

$y:=0,0\ 5\ \ 3$

x	g(x,3)
0	9
0 5	10 5
1	12
1 5	13 5
2	15

y	g(2,y)
0	0
0 5	2 5
1	5
1 5	7 5
2	10
2 5	12.5
3	15

说明：

(1)本例中 g 函数取自例 3 。

(2)前两行定义自变量序列。在此，0 5 是步长；“ ”可由键盘上的分号“;”获得。

(3)第三行是计算指令。它的输入方式是：键入(引号中的)“x=”，在其下方立即生成一个序列，然后按空格键，再键入“g(x,3)=”，得第二列，依次类推。

【例 5】定积分运算。

$$\int_0^1 \frac{1}{1+x^2}dx = 0\ 785$$

【例 6】输入矩阵 A，并求逆。

$$A = \begin{pmatrix} 4 & 5 & 1 \\ 5 & 0 & -12 \\ -7 & 2 & 8 \end{pmatrix}$$

$$A^{-1} = \begin{pmatrix} 0\ 074 & -0\ 117 & -0\ 184 \\ 0\ 135 & 0\ 12 & 0\ 163 \\ 0\ 031 & -0\ 132 & -0\ 077 \end{pmatrix}$$

说明：

(1)矩阵输入方法：单击操作板中的矩阵图标(或在【Math】菜单中选择【Matrices】)；选择列维与行维；填入元素。

(2)得到 A 阵逆的操作方法是，依次键入“A”、“^”、“-1”、“=”。

【例 7】二维曲线(图 1 4-2)。

$a\ = 4$

$$f(x)\ = \frac{\cos(x)}{a+\frac{x}{2}}$$

$x\ = 0,\ 1\ \ 25$

说明：

(1)Mathcad 的作图步骤：先定义自变量的取值区间；在图形操作板上选取 X－Y 绘图按钮或键入符号“@”绘图操作符，便出现一张空白图纸；在横轴中间的黑点处，键入自变量名；在

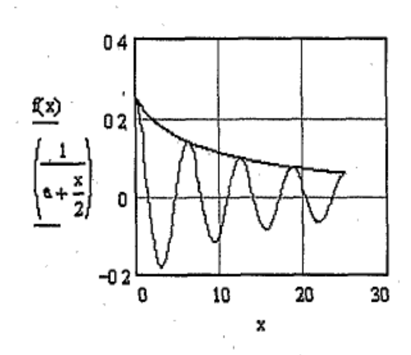


图 1.4-2 Mathcad 绘制的图形

纵轴中间黑点处，键入所需绘制的表达式或(和)已定义函数名，彼此间用逗号分开，就得相应的图形。

(2)绘完后，若改变参数、函数或表达式，则可立即看到 x 区间的重新定义以及图形的重新绘制。

第二章　MATLAB 的基础准备及入门

本章有两个目的：一是讲述 MATLAB 正常运行所必须具备的基础条件；二是帮助新用户在领略 MATLAB 非凡能力的同时比较轻松地跨过 MATLAB 门槛。

本章内容基本上按照进入 MATLAB 的前后次序编排。前三节分别讲述：运行 MATLAB 所需的外部环境、正确的安装方法、MATLAB 的目录结构和环境变量。第 2.4 节以较通俗的叙述、算例和在线演示帮助用户学习和掌握 MATLAB 的基本知识。鉴于 MATLAB 本身配有很好的在线查询系统，也鉴于用户愈早掌握它们得益也就愈多的思考，在第 2.5 节对 MATLAB 软件的在线查询系统作比较系统地讲述。本章的最后一节着重讲述如何扩展 MATLAB 搜索路径，以解决工作环境与用户工作目录或其他目录之间的信息交换问题。

2.1　对外部系统的要求

MATLAB 只有在适当的外部环境中才能正常运行。因此，恰当地配置外部系统是保证 MATLAB 运行良好的先决条件。MATLAB 本身可适应于许多机种和系统，如 IBM-PC、Macintosh 和 Unix 工作站等。但本节只针对我国使用最广的 PC 机系统给予介绍。

2.1.1　MATLAB 3.0、3.5 版对系统的要求

硬件要求：

(1) IBM-PC XT 以上机型，80386 及其以下机型必须配置相应的数学协处理器(8087、80287、80387)；

(2) 至少 320KB 的系统内存，推荐使用 1MB；

(3) 可以使用 EGA 卡，建议使用 VGA 卡；

(4) 至少需要 1.2MB 的硬盘空间。

软件要求：DOS 2.0 以上。

2.1.2　MATLAB 4.0、4.1、4.2 版对系统的要求

硬件要求：

(1) 基于 IBM-PC 的 80286，80386，80486 或奔腾的各种机型，80386 及其以下机型必须有相应的数学协处理器；

(2) 至少 4MB 的系统内存，推荐使用 8MB 以上；

(3) VGA 以上的彩显卡；

(4) 鼠标(Mouse)虽非必需，但为了能在 Windows 下工作轻松而迅捷，建议使用；

(5) 至少16MB的硬盘空间(不包括工具箱Toolbox时);

(6) 声卡(有少量指令需要)。

软件要求:

(1) MS-DOS或PC-DOS 3 3版以上的操作系统,以及Microsoft Windows 3 x或Microsoft Windows 95的中英文版;

(2) 当使用Notebook时,还需要MS-Word 6 0或以上的中英文版(MATLAB 4 0版不能使用Notebook)。

说明:当使用MATLAB Notebook时,由于同时运行MATLAB和Word,计算机的物理内存以8MB以上为宜。

2.2 MATLAB的安装

对在PC机上使用MATLAB的用户来说,常常需要自己安装和维护MATLAB。4 0版以前的MATLAB安装都比较简单,本书不再介绍。由于4 0及以后版本的安装方法大体上相同,所以本节将以4 2版为例较详细地介绍几种安装方法。

本书建议用户尽量使用标准安装方法,以确保MATLAB可靠运行。但有关非标准安装的内容定能帮助用户更好地维护MATLAB。

2.2.1 标准安装

MATLAB 4 2版核心部分(不含工具箱)安装程序以压缩形式(约6 4MB)存储在软盘上。MATLAB必须在Windows环境下安装。其安装方法有好几种,下面介绍一种常用方法。

(1) 启动Windows。

(2) 打开程序管理器(Program manager)主群组(Main)的文件管理器(File manager)。

(3) 若MATLAB的安装盘在A驱动器中,那么选择文件管理器窗口中的A驱动器图标,于是就出现如图2 2-1所示的A盘目录。

(4) 用鼠标双击setup exe,进入MATLAB的安装过程。

(5) 依照MATLAB安装程序提示的步骤,依次放入MATLAB的程序盘。需要说明的是:

第一,安装程序会提示用户选择MATLAB的工作目录;如果用户不提供,那么将自动定义缺省工作目录为C:\MATLAB;

第二,安装程序会提示用户选择安装方案,如果选择全部安装,将需16MB硬盘空间(对4 2版);

第三,安装程序还会提示用户安装工具箱(Toolbox)。如没有则略过;如有,则需要更多的硬盘空间。

(6) 安装的最后,程序会自动创建如图2.2-2所示的MATLAB FOR WINDOWS的程序组。此后,MATLAB就可以很方便地使用了。

安装程序执行完后,会在指定的硬盘驱动器上建立MATLAB的目录结构,无需用户修改

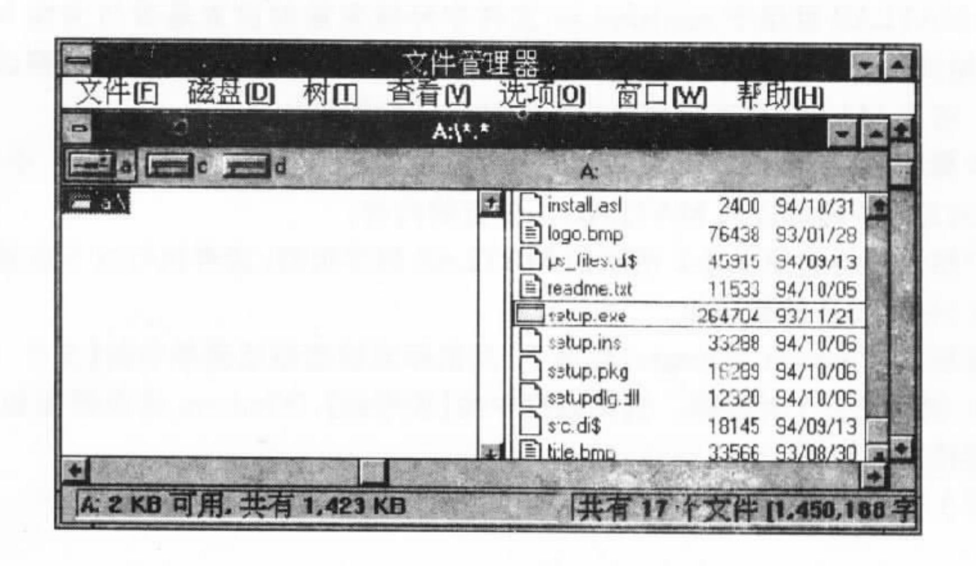


图 2.2-1 文件管理器打开的 A 盘目录

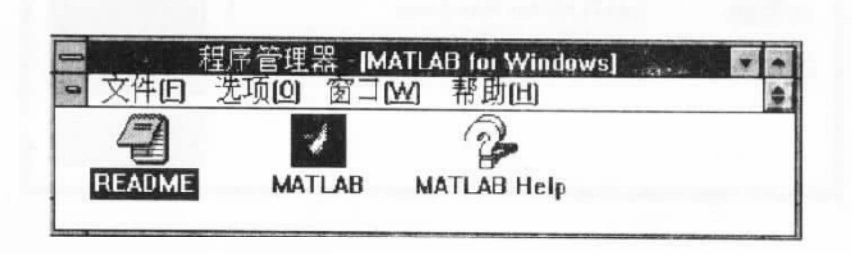


图 2.2-2 MATLAB for Windows 的程序组

就能直接运行 MATLAB。

说明:

(1) 当 MATLAB for Windows 的程序组(如图 2.2-2)建立后,用户只需用鼠标点击 MATLAB 运行文件的图标就可以运行 MATLAB。

(2) MATLAB 安装完后,在 Windows 的工作目录下建立两个文件:

matlab.ini 是 Windows 启动 MATLAB 的初始化文件。该文件定义了 MATLAB 启动时的初始参数。

matlabfo.grp 是 Windows 在程序管理器(Programm manager)中建立 MATLAB 程序组的组群文件。

2.2.2 非标准安装

若 MATLAB 原安装盘受到损坏,或者已建立的 MATLAB 的完整性遭到破坏以及需要进行不同机器间的 MATLAB 转移,那么非标准安装也许是必要的。

非标准安装的步骤如下:

(1)要准确保存和记录原 MATLAB 的目录结构、名称及文件。

(2)在待安装的目标驱动器上建立 MATLAB 目录,然后将文件按原目录结构及名称逐一拷入。

(3)检查 MATLAB 目录下 matlabrc.m 文件中环境变量的设置是否与当前 MATLAB 所在的盘、目录结构、名称一致。假如不同，就需要对由 matlabpath 的内容进行修改，使它与当前情况相符。有关 MATLAB 环境变量设置的细节请参阅第 2.3 节。

在以上步骤结束后，MATLAB 就可以运行了。具体运行方法，见第 2.4.1 小节中关于利用 Windows 的文件管理器进入 MATLAB 工作窗的内容。

假如用户想进而构造图 2.2-2 所示的 MATLAB 程序组群，请再执行以下步骤：

(1) 建立 MATLAB 的程序组

在程序管理器(Program Manager)环境下，用鼠标或键盘激活菜单中的【文件[F]】选项，选择【新建[N]】，便弹出一个对话框。选择该框中的【程序组】，Windows 进而弹出如图 2.2-3 所示的程序组特性对话框。

按图 2.2-3 所示，填写程序组的说明和组文件的名字，选择【确定】键。

程序组特性

说明(D):　MATLAB for Windows　　确定

组文件(G):　C:\WINDOWS\MATLABFO.GRP　　取消

帮助(H)

图 2.2-3　MATLAB 程序组特性

(2) 建立 MATLAB 运行文件程序项

在程序管理器(Program Manager)环境下，用鼠标或键盘激活菜单中的【文件[F]】选项，选择【新建[N]】。然后选择【程序项】。

当程序项对话框弹出后，按图 2.2-4 填写各栏内容。

点击【更改图标[I]】键，选择图标。然后点击【确定】键。

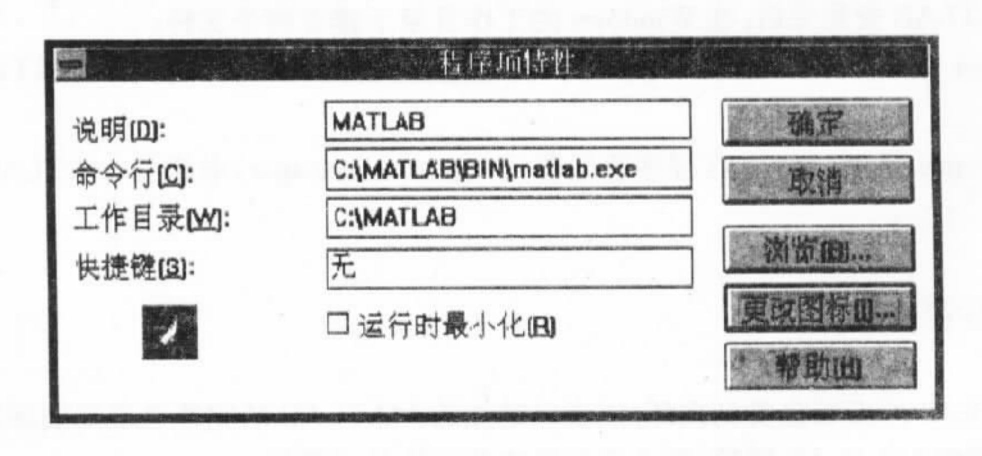


图 2.2-4　MATLAB 运行文件程序项

(3) 建立 README 程序项

整个过程与前相同。填写内容如图 2.2-5 。

注意，命令行的完整内容是：NOTEPAD.EXE C:\MATLAB\README.TXT。



图 2.2-5 MATLAB README 程序项特性

(4) 建立 MATLAB help 程序项

整个过程与前面相同。填写内容如图 2.2-6 。

注意,命令行的完整内容是:WINHELP C:\MATLAB\BIN\MATLAB.HLP。



图 2.2-6 MATLAB Help 程序项特性

说明:

(1) 以上设置,都假定 MATLAB 保存在 C 盘上;否则,在设置过程中要取相应盘名代替 C。

(2) 整个程序组群创建完成后的结果与标准安装完全相同。

2.3 MATLAB 的目录结构与环境变量

本节内容能帮助用户进一步理解和掌握 MATLAB,但初学者可以跳过本节。

2.3.1 目录结构

MATLAB 安装结束后,在硬盘上形成一个 MATLAB 目录(比如 C:\MATLAB)。在这目录中会有如下文件和子目录:

matlabrc.m

readme.txt

\ bin

\ extern

\ ghostscr

\ notebook

\ simulink

\ toolbox

说明：当只安装 MATLAB 核心软件时，就没有 \ notebook 和 \ simulink 。

(1) matlab \ bin

该目录包含 MATLAB 系统运行文件，建立 MEX-文件所需批处理文件，MATLAB 帮助文件及一些必需的二进制文件。

(2) matlab \ extern

MATLAB 与 C，FORTRAN 语言的交互所需的函数定义和链接库。它有三个子目录：

\ include　　C 语言源程序的头文件(include 文件)，用于变量及函数的定义。

\ lib　　外部接口的目标链接库。

\ src　　一些 C 和 FORTRAN 源程序举例。

(3) matlab \ notebook

在仅安装 MATLAB 主软件的情况下，不存在 notebook 子目录。Notebook 是 MathWorks 公司为 MATLAB 配套使用而提供的附属功能软件，它需要专门安装。

该子目录包含了实现 MATLAB 数学工作环境与 MS-Word 文字处理环境信息交换所必需的软件(M-book.dot 模板)和一些示范文件。该目录的存在将为用户提供一个兼备数学计算、图形显示、文字处理能力的集成环境。关于 Notebook 的作用及使用方法将在第六章讨论。

(4) matlab \ simulink

本目录包含建立 Simulink MEX-文件所必需的函数定义及接口软件。

Simulink 是一个专门用于动态系统仿真分析的软件环境。在 MATLAB 4.0 版，Simulink 功能包含在 matlab-s.exe 中。而对 MATLAB 4.1、4.2 版，Simulink 是可选的附属功能软件，即用户必须另外购买 MATLAB 的 Simulink，然后安装到 MATLAB 中去。

(5) matlab \ toolbox

它包含以下子目录：

\ matlab　　MATLAB 核心部分工具包。

\ simulink　　Simulink 函数和工具。

\ symbolic　　实现符号计算的 M-文件和 Maple 的核与库。

\ wintools　　MS-Windows 环境下图形用户界面工具。

说明：当只安装 MATLAB 核心软件时，在 toolbox 目录下只有 \ matlab 和 \ wintools 。到目前为止，MathWorks 公司提供的商品化 MATLAB 应用工具包有三十来种。因此，toolbox 目录下子目录的内容和数量随用户的安装情况而变。

(6) matlab \ ghostscr

解释和运行页面描述文件(Postscript)的共享性软件组，用以产生专业印刷质量的图形硬拷贝。

(7) matlab \ readme.txt

MATLAB 版本最新更改信息。它包括用户手册上没罗列的最新版本信息、特点、新版对系统的要求、硬件要求和已发现的新版设计缺陷。

(8) matlab \ matlabrc m

MATLAB 启动时所执行的文件，详见下节。

2.3.2 MATLAB 环境变量

MATLAB 环境变量由 matlabrc m 定义。matlabrc m 是在 MATLAB 启动后自动执行的一个 m-文件。它定义了 MATLAB 环境下的路径结构、MATLAB 图形的大小、图元缺省值和 MATLAB 工作窗口的初始提示信息等重要参数。该文件定义的搜索路径部分如下：

```
matlabpath([
';C:\MATLAB\toolbox\local',
';C:\MATLAB\toolbox\matlab\datafun',
';C:\MATLAB\toolbox\matlab\elfun',
';C:\MATLAB\toolbox\matlab\elmat',
';C:\MATLAB\toolbox\matlab\funfun',
';C:\MATLAB\toolbox\matlab\general',
';C:\MATLAB\toolbox\matlab\color',
';C:\MATLAB\toolbox\matlab\graphics',
';C:\MATLAB\toolbox\matlab\iofun',
';C:\MATLAB\toolbox\matlab\lang',
';C:\MATLAB\toolbox\matlab\matfun',
';C:\MATLAB\toolbox\matlab\ops',
';C:\MATLAB\toolbox\matlab\plotxy',
';C:\MATLAB\toolbox\matlab\plotxyz',
';C:\MATLAB\toolbox\matlab\polyfun',
';C:\MATLAB\toolbox\matlab\sounds',
';C:\MATLAB\toolbox\matlab\sparfun',
';C:\MATLAB\toolbox\matlab\specfun',
';C:\MATLAB\toolbox\matlab\specmat',
';C:\MATLAB\toolbox\matlab\strfun',
';C:\MATLAB\toolbox\matlab\dde',
';C:\MATLAB\toolbox\matlab\demos',
';C:\MATLAB\toolbox\simulink\simulink',
';C:\MATLAB\toolbox\simulink\simdemos',
';C:\MATLAB\toolbox\simulink\blocks',
';C:\MATLAB\toolbox\simulink\sb2sl',
';C:\MATLAB\toolbox\symbolic',
';C:\MATLAB\toolbox\symbolic\lib',
';C:\MATLAB\toolbox\wintools',
]);
```

MATLAB 的搜索路径比较复杂。上面所列的搜索目录只包括 MATLAB 的基本部分、

Simulink、Symbolic，还不包含MATLAB的其他应用工具包。

当MATLAB正常安装时，该路径自动生成，不需用户编写。故此，本文向用户推荐使用"MATLAB的标准安装法"；并提醒不熟悉MATLAB结构的用户，不要随便改动此文件，否则可能导致MATLAB工作失常。

但matlabpath也并不神秘，它实际上是MATLAB的一个指令：定义了MATLAB系统所能搜索的范围(目录及子目录)，定义了搜索变量或函数时所循的路径和前后次序。

在MATLAB中，如果键入一个命令，如example，那么MATLAB将进行以下搜索：

(1) 在工作内存中搜索，看它是否是变量。

(2) 然后，再检查它是否是内部函数。

(3) 在当前目录下，检查是否有example.m文件。

(4) 沿matlabrc m指定的路径，在逐个目录中寻找是否有example m文件。

matlabrc m是一个标准的M文件，当用户熟悉了MATLAB以后，可以对它进行修改，也可以将用户自己的工作目录加到matlabrc m之中。MATLAB 4 2版还提供了一个专门的路径修改命令pathtool。关于pathtool指令，将在第2 6 2节和第六章中介绍。

2.4 MATLAB入门

本节旨在引导初学者进入MATLAB工作环境，帮助他们初步了解MATLAB最基本的操作方法，让他们具备MATLAB最基本的工作能力。对于初学者来说，若能循本节所给算例实际操作一遍，就不难进入MATLAB之门。

2.4.1 MATLAB的启动

本节向初学者介绍进入MATLAB工作环境的方法。

利用图标进入MATLAB工作窗口

进入MATLAB工作窗口最基本、最容易的方法是利用图2.2-2所示的MATLAB图标。具体步骤如下：

(1) 进入Windows；

(2) 打开程序管理器(Program manager)的MATLAB for Windows程序组；

(3) 用鼠标双击MATLAB图标，启动MATLAB出现一个如图2.4-1所示的指令窗MATLAB Command Window。

说明：

(1) MATLAB工作窗最上方的两行文字是初始提示信息。

(2) 若MATLAB运行在英文Windows平台上，那么MATLAB工作窗中的第三行将出现MATLAB环境提示符号"》"和光标位置符。

(3)在中文Windows平台上的MATLAB工作窗中，将不显示提示符号"》"，而只有光标位置符。

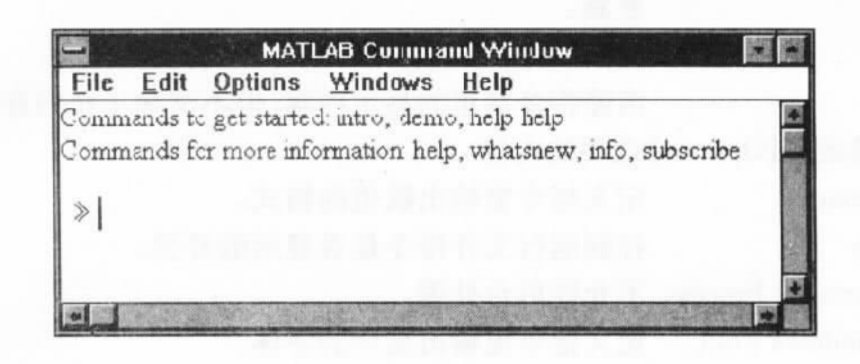


图 2.4-1 MATLAB 的指令窗

利用文件管理器(File manager)进入 MATLAB 工作窗

这种启动方式不如前种方法直观,但也相当方便。具体办法是:

(1) 进入 Windows ;

(2) 打开程序管理器主群组(Main)中的文件管理器;

(3) (若 MATLAB 安装在 C 盘上)通过鼠标操作依次进入 C 盘、matlab 目录、bin 子目录;

(4) 用鼠标双击 matlab. exe 文件,启动 MATLAB,也可得到图 2.4-1 所示的 MATLAB 工作窗。

2.4.2 工作窗和指令行的操作

在 MATLAB 工作窗出现以后,用户就可以在工作窗里进行各种运算操作。为此,本节要介绍该工作窗的基本环境和指令行的基本操作。

菜单选项

MATLAB 工作窗是标准的 Windows 工作界面,因此可以利用工作菜单中的各种选项来实现对工作窗中内容的操作。它的使用方法也是标准的。下面将逐项给予介绍。

(1) 基本文件操作【File】选项的内容

New　　建立新的 M-文件,图形或 Simulink 的模块。

Open M-File　　打开已经存在的 M-文件。

Open Selected　　打开指令窗中指定的 M-文件。

Save Workspace As　　将 MATLAB 工作空间中的内容存入文件。

Run M-file　　运行已有的 M-文件。

Lookfor Selected　　搜寻工作空间中的指定文件。

Print　　打印工作窗中的内容。

Print Setup　　打印设置。

Exit MATLAB　　退出 MATLAB。

(2) 编辑操作【Edit】选项的内容

Cut　　剪切。

Copy 复制。

Paste 粘贴。

Clear Session 清除指令窗里的显示内容，但不清除工作内存中的变量。

(3) 工作环境定义【Options】选项的内容

Numeric Format 定义指令窗输出数值的格式。

Turn Echo on 控制运行文件指令是否显示的开关。

Disable Background Process 不允许后台处理。

Command Windows Font 定义指令窗输出显示的字体。

Uicontrols Font 定义图形控制按钮的字体。

Edit Preference 选定 M-文件的编辑器。

(4) MATLAB 环境下工作窗管理【Windows】选项

如果没有图形的话，则只有一个【1 MATLAB Command Window】选项；如果有图形的话，则会有相应的图形窗选项。

(5) 帮助【Help】选项内容

Table of Contents 帮助文件的分类索引表。

Index 按字母次序排列的指令索引表。

Help Selected 对选定内容的帮助。

About 关于 MATLAB。

若干通用操作指令

除了通过菜单选项对工作窗进行控制外，MATLAB 还提供了许多通过键盘输入的控制指令。MATLAB 工作窗中的一些通用操作指令见表 2.4-1。

表 2.4-1 MATLAB 工作窗中的部分通用指令

quit	关闭和退出 MATLAB
clc	擦除 MATLAB 工作窗中的所有显示内容
clf	擦除 MATLAB 的当前图形窗中的图形
clear	清除内存中的变量和函数
pack	收集内存碎块以扩大内存空间
dir	列出指定目录下的文件和子目录清单
cd	改变当前工作子目录
disp	(在运行中)显示变量或文字内容
type	显示所指定文件的全部内容
echo	控制运行文件指令是否显示的开关
hold	控制当前图形窗对象是否被刷新

指令行的编辑

启动 MATLAB 后，就可以利用 MATLAB 工作。由于 MATLAB 是一种交互式语言，随时输入指令、即时给出运算结果是它的主要工作方式之一。

比方说，用户想计算$\frac{2\sin(0.3\pi)}{1+\sqrt{5}}$的值，那么应在光标位置处依次键入以下字符：

```
2 * sin(0 3 * pi)/(1+sqrt(5))
```

然后按【Enter】键，该指令便被执行并给出以下结果：

```
ans =
    0.5000
```

在此，不介绍 MATLAB 究竟有些什么函数，也不准备介绍 MATLAB 究竟有些什么内部变量，而主要介绍控制光标位置、对指令行进行编辑的一些常用操作键，见表 2.4-2。

表 2.4-2　常用操作键

键　名	作　用	键　名	作　用
↑	前寻式调回已输入过的指令行	Home	使光标移到当前行的首端
↓	后寻式调回已输入过的指令行	End	使光标移到当前行的尾端
←	在当前行中左移光标	Delete	删去光标右边的字符
→	在当前行中右移光标	Backspace	删去光标左边的字符
PageUp	前寻式翻阅当前窗中的内容	Esc	清除当前行的全部内容
PageDown	后寻式翻阅当前窗中的内容		

比如，用户又想计算$\frac{2\cos(0\ 3\pi)}{1+\sqrt{5}}$，可以像前一个算例那样，通过键盘把相应字符一个一个"敲入"。但也可以较方便地用操作键获得该指令：先用↑键调回已输入过的指令 2 * sin(0 3 * pi)/(1+sqrt(5))，然后移动光标、把 sin 改成 cos 便可。

本书约定

（1）每个指令或指令行键入后，都必须按【Enter】键，只有这样输入的指令才会被执行。但为简炼起见，将不再重复叙述"按【Enter】键"。

（2）凡是 MATLAB 工作窗中的实际运行指令，都用"方头黑体"英文及数字表示；运算结果用"细体"英文及数字表示。而一般叙述中的英文及数字采用"白正体"字体。

（3）由于本书的全部内容是用 MATLAB 的 Notebook 写成，因此指令行前都没有 MATLAB 环境提示符号"》"。

（4）由于篇幅原因，有些例题的图形输出并没有排在相应指令的下方。

2.4.3　工作窗中提示信息简介

本节着重介绍 MATLAB 工作窗第一提示行中的三个指令：intro，demo，help help。假如用户能把这三个命令试运行一遍，定能从中获得颇为丰富的收益。

在线提供的"入门演示"

初学 MATLAB 的用户，将从 intro 指令的运行中，学到 MATLAB 最基本的使用方法，并看到 MATLAB 的灵活、方便和强大的功能。有两种方法可看到 MATLAB 在线提供的"入门演示"，在本段落中只介绍一种，具体是在 MATLAB 工作窗中，运行以下指令：

intro

该指令运行后，先出现一幅 MATLAB 彩色图标，按任意键便出现"入门演示"的实质性内

容，以后每按一次任意键，就更新一组内容。

另一种获得"入门演示"的方法在下一段落中配合 demo 指令一起介绍。

在线演示

MathWorks 公司精心设计了一组旨在介绍 MATLAB 功能的演示程序。运行这组程序，对照屏幕上的显示仔细研究实现演示的有关 M 文件，无论是对 MATLAB 新用户来说还是对老用户来说，都是十分有益的。该演示程序的示范作用是独特的，是包括 MATLAB 用户指南在内的有关书籍所不能代替的。用户若想学习和掌握 MATLAB，不可不看这组演示程序。

但对初学者来说，不必急于去读那些太复杂的程序。下面介绍怎样用 demo 指令看"入门演示"，步骤是：在 MATLAB 工作窗中，运行以下指令：

demo

该指令运行后，打开一个"Welcome to the MATLAB Expo !"窗口，待 MATLAB 图标旋转结束，再按【Continue】按钮，便打开"Expo Map Windows"窗口下的主菜单"The MATLAB Expo MAIN MAP"；选择 MATLAB 栏，点击【Visit】按钮，又进入分菜单"MATLAB"；点击"MATRICES"分栏下的【Select a demo】按钮，出现菜单细目表；再点击"intro"行，便出现图 2.4-2 所示的"Slide Show Player"窗口。

为能逐页观察"入门演示"，请用户在"Slide Show Player"窗口中，先点击【Start】按钮，再点击【Next】按钮。此后，就出现如图 2.4-3 所示的"入门演示"内容。以后，每按一次【Next】按钮，就打开下一页，直至结束。

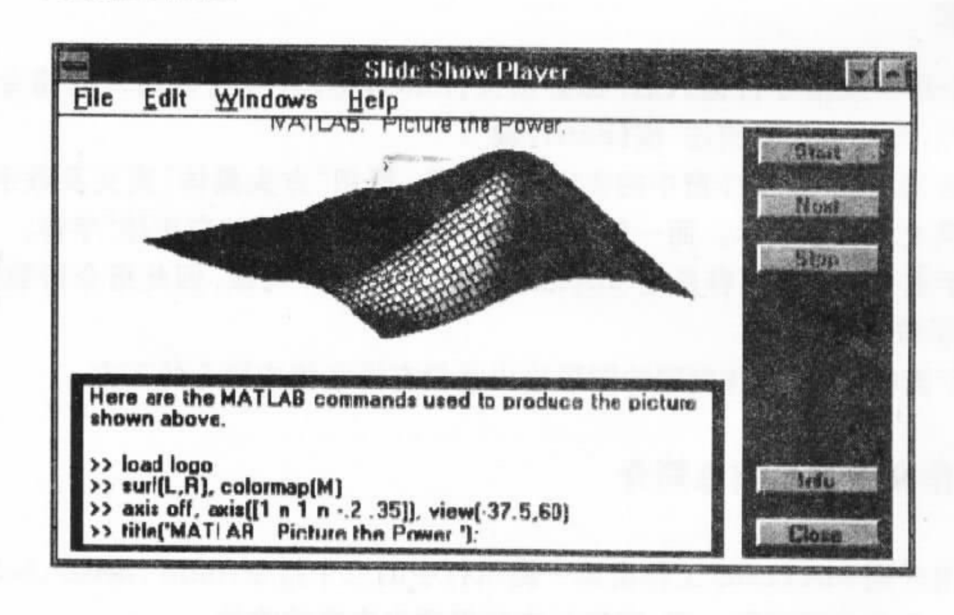


图 2.4-2　MATLAB 图标及其生成指令

关于图 2.4-2 的说明：该窗口下面的方框中，给出了生成 MATLAB 图标的四行指令。

第一行是从数据文件 logo.mat 中加载外部数据，获得变量 L、R、M、n 。

第二行的前一个指令画三维曲面图，后一个指令设置曲面的颜色。

第三行的三个指令先后实现：隐藏坐标轴，设定坐标范围，选定观察视角。

第四行赋图形以名称（在中文 Windows 上，也可以给中文名）。

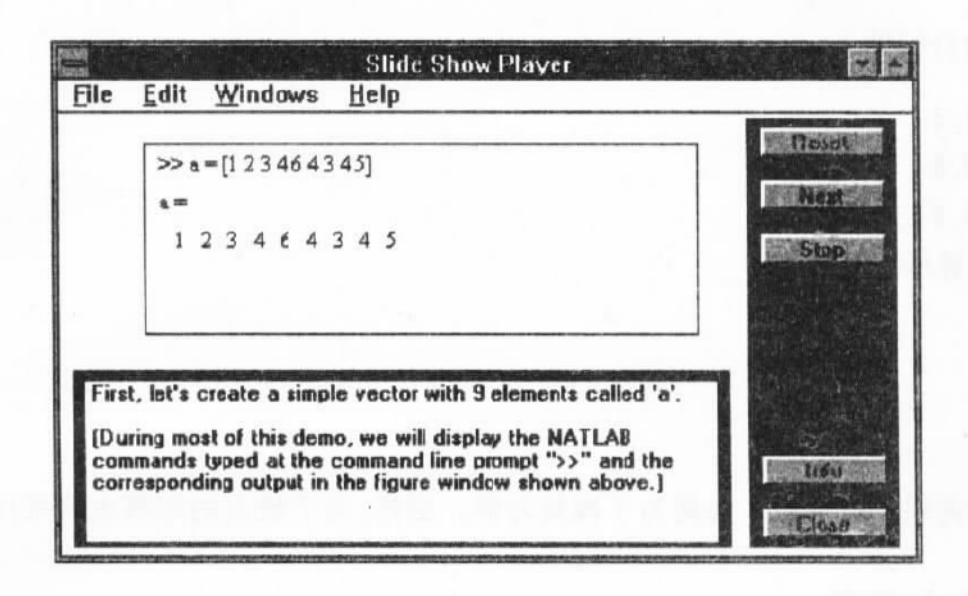


图 2.4-3 "入门演示"内容的第一页

<u>关于在线帮助的入门提示</u>

help help 指令的运行将告诉用户:在 MATLAB 中,有哪些在线帮助指令及如何利用它们(关于 help 和 lookfor 的使用,将在第 2.5 节详述)。

2.4.4 简单矩阵的输入

在 MATLAB 中,输入矩阵的方法有好多种。本节只简单介绍矩阵的直接输入法,详细介绍将在第三章进行。

在 MATLAB 中,不必对矩阵维数做任何说明,存贮将自动配置。在直接输入矩阵时,矩阵元素用空格或逗号分隔,矩阵行用分号";"隔离。整个矩阵放在方括号"[ ]"里。

【例 1】简单矩阵的输入步骤。

(1) 在键盘上输入下列内容:

```
A = [1,2,3; 4,5,6; 7,8,9]
```

(2) 按【Enter】键,指令被执行。

(3) 在指令执行后,MATLAB 指令窗中将显示以下结果:

```
A =
     1     2     3
     4     5     6
     7     8     9
```

说明:指令执行后,矩阵 A 被保存在 MATLAB 的工作间(Workspace)中,以备后用。如果用户不用 clear 指令清除它,或对它重新赋值,那么该矩阵会一直保存在工作间中,直到本 MATLAB 指令窗被关闭为止。

【例 2】矩阵的分行输入。

```
A=[1,2,3
   4,5,6
   7,8,9]
```

(以下是显示结果)

```
A =
     1     2     3
     4     5     6
     7     8     9
```

说明:本例采用这种输入法是为了视觉习惯。当然,对于较大的矩阵也可采用此法。

2.4.5 语句与变量

MATLAB 采用表达式语言。用户输入的语句由 MATLAB 系统解释运行。MATLAB 语句有两种最常见的形式:

(1) 表达式

(2) 变量 = 表达式

说明:

(1) 表达式由算符、函数、变量名和数字构成。

(2) 在第一种形式中,表达式被执行后产生的矩阵,将被自动赋给名为“ans”的变量,并显示在屏幕上。“ans”是一个缺省变量名,它会被以后类似的操作刷新。

(3) 在第二种形式中,等号右边的表达式是被演绎后产生的矩阵,将被赋给等号左边的变量后放入内存,并显示在屏幕上。

(4) 书写表达式时,运算符号“=”、“+”、“-”以及“*”等两侧允许有空格,以增加可读性。但在复数或符号表达式中,要尽量避免“装饰性”空格,以防出错。

(5) 变量名、函数名以一个字母打头,后面最多可接 19 个字母或数字。注意,MATLAB 是区分字母的大小写的。

【例 1】表达式的计算结果。

```
1996/18
ans =
   110.8889
```

【例 2】运算结果的赋值。

```
S=1-1/2+1/3-1/4+1/5-1/6-...
    1/7+1/8 ;
```

说明:

(1) 本语句第一行结尾的三个小黑点是“续行号”,它表示下一行是上一行的继续。

(2) 本语句第二行结尾的分号“;”作用是:指令执行结果将不显示在屏幕上,但变量 S 仍将驻留在内存中。若用户想看 S 的值,可键入以下指令:

```
S
```

(以下是显示结果)

```
S =
    0 5988
```

2.4.6 Who、Whos 和永久变量

who 和 whos

who 和 whos 这两个指令的作用都是列出在 MATLAB 工作间中已经驻留的变量名清单。不过,whos 在给出驻留变量名的同时,还给出它们的维数及性质。

【例 1】用 who 检查内存变量。

```
who
```

(以下是显示内容)

```
Your variables are:
A        S              ans
```

说明:

(1) 获得本例显示内容的前提是:自第二章以来所开设的 MATLAB 指令窗中,用户是按本章算例实际操作的。

(2) 显示结果表示:在前面的举例中已经产生了 A, ans ,S 三个变量。

【例 2】键入 whos ,获得这里驻留变量的详细情况。

```
whos
```

(以下是显示内容)

Name	Size	Elements	Bytes	Density	Complex
A	3 by 3	9	72	Full	No
ans	1 by 1	1	8	Full	No
S	1 by 1	1	8	Full	No

Grand total is 10 elements using 80 bytes

永久变量

在 MATLAB 工作内存中,还驻留几个由系统本身在启动时定义的变量见表 2.4-3。MATLAB 用户手册把它们称为永久变量(Permanent variables)。有一些关于 MATLAB 的文献则把它们称为预定义变量(Predefined variables)。这些变量有以下特点:

表 2.4-3 系统启动时定义的变量

eps	容差变量，定义为 1 0 到最近浮点数的距离。在 PC 机上，它等于 2^{-52}
pi	圆周率 π 的近似值 3 14159265358979
inf 或 Inf	正无穷大，定义为$(\frac{1}{0})$
NaN	非数(Not a number)。在 IEEE 运算规则中，它产生于$\frac{0}{0}$，$\frac{\infty}{\infty}$，$0\times\infty$等运算
i，j	虚数单位，定义 $i=\sqrt{-1}$，$j=\sqrt{-1}$

(1) 它们是在 MATLAB 启动时自定义的。

(2) 它们不会被"清除内存变量"指令 clear 所清除(永久变量的名称就源于此)。

(3) 它们可以重新定义为其他值，但用 clear 可清除重定义值，恢复预定义值(预定义变量的名称就反映着这一层含义)。

(4) 在 MATLAB 3 0 版中，可以用 who 指令查看这些永久变量。但在 MATLAB 3 5、4 0、4 1 和 4 2 版中都不能用 who 指令查看到永久变量。

【例 3】无穷大。

```
s = 1/0
Warning: Divide by zero
s =
    Inf
```

说明：在带 IEEE 算法的机器上，被 0 除是允许的。它不会导致程序执行的中断，只是在给出警告信息的同时，用一个特殊名称记述。这个特殊名称将在以后的计算中以合理的形式发挥作用。

2.4.7 数与表达式

数的记述

MATLAB 的数值采用习惯的十进制表示，可以带小数点或负号。以下记述的数都是合法的：

3	−99	0 001
9 456	1 3e−3	4 5e33

在采用 IEEE 浮点算法的计算机上，数值的相对精度是 eps，即大约保持 16 位有效数字。数值范围大致为 $1\times10^{-308}\sim1\times10^{308}$。

表达式的构成

表达式由下列算符构成，并按习惯的优先次序进行运算。

\+ 加法　　− 减法

*　乘法　　/　右除　　\　左除

^　幂

注意：设置两种除法是为了方便矩阵的运算。对标量来说，两者作用相同，详见第三章。

【例 1】运算。

```
x=2*pi/3+2^3/5-0.3e-3
x =
    3.6941
```

2.4.8　复数和复矩阵

MATLAB 认识复数，并用预定义变量 i 和 j 作为虚数单位。MATLAB 的矩阵元素允许是复数、复变量和由它们组成的表达式。

【例 1】复数表达及运算。

```
z1=3+4*i,z2=2*exp(i*pi/6)
z=z1*z2
z1 =
     3.0000 + 4.0000i
z2 =
     1.7321 + 1.0000i
z =
     1.1962 + 9.9282i
```

【例 2】复数矩阵的生成及运算。

```
A=[1,3;2,4]-i*[5,8;6,9]
B=[1+5*i,2+6*i;3+8*i,4+9*i]
c=A*B
A =
    1.0000 - 5.0000i    3.0000 - 8.0000i
    2.0000 - 6.0000i    4.0000 - 9.0000i
B =
    1.0000 + 5.0000i    2.0000 + 6.0000i
    3.0000 + 8.0000i    4.0000 + 9.0000i
c =
    1.0e+002 *
    0.9900              1.1600 - 0.0900i
    1.1600 + 0.0900i    1.3700
```

2.4.9 图 形

图形是 MATLAB 主要特色之一。MATLAB 图形指令具有自然、简洁、灵活及易扩充的特点。用户一旦试过 MATLAB 作图指令，就很难放弃这种图形功能。MATLAB 的图形指令很多，这里只介绍几个简单的绘图指令，详见第五章。

【例 1】作多条曲线(图 2 4-4)。

```
t=0:pi/50:4*pi;
y0=exp(-t/3);
y=exp(-t/3).*sin(3*t);
plot(t,y,'-r',t,y0,'--b',t,-y0,'--b')
grid
```

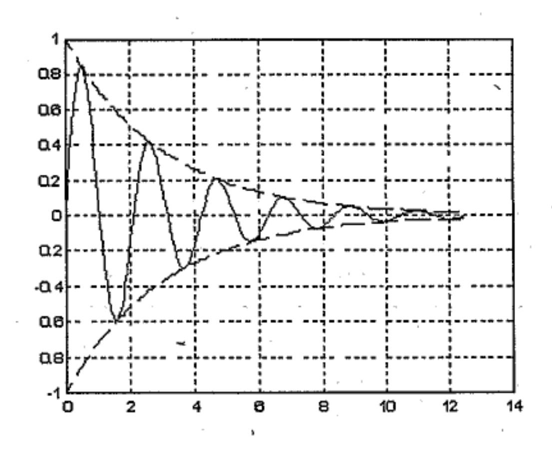


图 2 4-4 衰减振荡曲线与包络

【例 2】三维曲面(图 2.4-5)。

```
x=-8:0 5:8;
y=x';
X=ones(size(y))*x;
Y=y*ones(size(x));
R=sqrt(X.^2+Y ^2)+eps;
Z=sin(R)./R;
mesh(Z);
colormap([1,0,0])
```

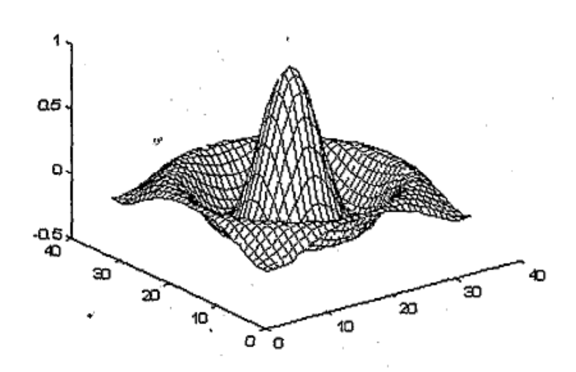


图2.4-5 三维网线图

2.5 MATLAB 的在线查询

理解、掌握和运用 MATLAB 的在线查询功能，对新老用户都是十分重要和必需的。MATLAB 的在线帮助系统相当完备。就查询系统的调用方式而言，可分两种：

一是从 MATLAB 指令窗的 help 菜单选项中寻求帮助。该菜单包括按字母排序的指令索引表和按内容排列的分类索引表。它的使用方法与一般 Windows 的求助方法一样，这里不再介绍。

二是在 MATLAB 指令窗内，直接键入帮助命令求助。这种方法最常使用。下面将较为详细地介绍这种求助方法。

2.5.1 help 指令

help 是最常用的求助指令。它可以提供绝大部分 MATLAB 指令使用方法的在线说明。本节将举例介绍 help 的应用方法。

【例 1】在线帮助总览。

```
help
(以下是显示内容)
HELP topics:
toolbox \ local        -    Local function library.
matlab \ datafun       -    Data analysis and Fourier transform functions.
matlab \ elfun         -    Elementary math functions.
matlab \ elmat         -    Elementary matrices and matrix manipulation.
matlab \ funfun        -    Function functions - nonlinear numerical methods.
matlab \ general       -    General purpose commands.
matlab \ color         -    Color control and lighting model functions.
```

```
matlab \ graphics     -  General purpose graphics functions.
matlab \ iofun        -  Low - level file I/O functions.
matlab \ lang         -  Language constructs and debugging.
matlab \ matfun       -  Matrix functions - numerical linear algebra.
matlab \ ops          -  Operators and special characters.
matlab \ plotxy       -  Two dimensional graphics.
matlab \ plotxyz      -  Three dimensional graphics.
matlab \ polyfun      -  Polynomial and interpolation functions.
matlab \ sounds       -  Sound processing functions.
matlab \ sparfun      -  Sparse matrix functions.
matlab \ specfun      -  Specialized math functions.
matlab \ specmat      -  Specialized matrices.
matlab \ strfun       -  Character string functions.
matlab \ dde          -  DDE Toolbox.
matlab \ demos        -  The MATLAB Expo and other demonstrations.
matlab \ simulink     -  (No table of contents file)
simulink \ simulink   -  SIMULINK model analysis and construction functions.
simulink \ simdemos   -  SIMULINK demonstrations and samples.
simulink \ sb2sl      -  SystemBuild 3.0 model import into SIMULINK.
simulink \ blocks     -  SIMULINK block library.
toolbox \ symbolic    -  (No table of contents file)
toolbox \ wintools    -  GUI tools for MATLAB for MS Windows.
For more help on directory/topic, type "help topic".
```

说明:如果想得到更详细的专题帮助,可在 help 指令后面再键入欲寻求的专题名字。

【例 2】寻求关于基本函数的帮助信息。

help elfun

(以下是显示结果)

```
Elementary math functions.

Trigonometric.
sin        - Sine.
sinh       - Hyperbolic sine.
asin       - Inverse sine.
asinh      - Inverse hyperbolic sine.
cos        - Cosine.
cosh       - Hyperbolic cosine.
acos       - Inverse cosine.
acosh      - Inverse hyperbolic cosine.
```

```
tan        - Tangent.
tanh       - Hyperbolic tangent.
atan       - Inverse tangent.
atan2      - Four quadrant inverse tangent.
atanh      - Inverse hyperbolic tangent.
sec        - Secant.
sech       - Hyperbolic secant.
asec       - Inverse secant.
asech      - Inverse hyperbolic secant.
csc        - Cosecant.
csch       - Hyperbolic cosecant.
acsc       - Inverse cosecant.
acsch      - Inverse hyperbolic cosecant.
cot        - Cotangent.
coth       - Hyperbolic cotangent.
acot       - Inverse cotangent.
acoth      - Inverse hyperbolic cotangent.

Exponential.
exp        - Exponential.
log        - Natural logarithm.
log10      - Common logarithm.
sqrt       - Square root.

Complex.
abs        - Absolute value.
angle      - Phase angle.
conj       - Complex conjugate.
imag       - Complex imaginary part.
real       - Complex real part.

Numeric.
fix        - Round towards zero.
floor      - Round towards minus infinity.
ceil       - Round towards plus infinity.
round      - Round towards nearest integer.
rem        - Remainder after division.
sign       - Signum function.
```

说明：若想进一步了解某指令的使用规则，可以在 help 后面，再键入那个指令名。

【例3】寻求指数函数指令 exp 的详细信息。

```
help exp
(以下是显示结果)
EXP Exponential.
    EXP(X) is the exponential of the elements of X, e to the X.

    See also LOG, LOG10, EXPM, ARITH, POW2.
```

说明：

(1) 在此显示的最后一行给出了与 exp 相关的指令。

(2) help 的工作机理是，把指定名字的那个 M 文件的第一段注释内容显示出来。用户采用这种注释结构，以构成自己文件的在线求助。

2.5.2 lookfor 指令

当要查找具有某种功能但又不知道准确名字的指令时，help 的能力就不够了。为此，MATLAB 设计了一个 lookfor 指令。它可以根据用户提供的完整或不完整的关键词，去搜索出一组与之有关的指令。

【例1】查找有关积分的指令。

```
lookfor integral
(以下是显示结果)
QUAD        Numerical evaluation of an integral, low order method.
QUAD8       Numerical evaluation of an integral, higher order method.
ELLIPK      Complete elliptic integrals.
ELLIPKE     Complete elliptic integrals.
EXPINT      Exponential integral function, E1(x).
FNINT       Indefinite integral of a function represented by splines.
```

【例2】寻找能进行傅里叶变换的有关指令。

```
lookfor fourier
(以下是显示结果)
contents.m: % Data analysis and Fourier transform functions.
FFT         Discrete Fourier transform.
FFT2        Two-dimensional Fast Fourier Transform.
IFFT        Inverse discrete Fourier transform.
IFFT2       Two-dimensional inverse discrete Fourier transform.
FOURIER     Graphics demo of Fourier series expansion.
```

XFOURIER Graphics demo of Fourier series expansion.
DFTMTX Discrete Fourier transform matrix.
EXPFOU Write data to a Fourier vector or a (maybe existing) file (for ELIS).
IMPFOU Read complex amplitudes from a Fourier vector or file (used by ELIS).
MODIFYFV Modify Fourier (maybe also variance) data by given transfer function.
SIMFOU Generate simulated Fourier amplitudes.
PLOTFOU Plot contents of Fourier files

说明:

(1) 由以上两个举例可以看出,lookfor 与 help 相配合就形成了相当完备的在线求助系统。

(2) lookfor 的机制是,对 MATLAB 目录中的每个 M 文件注释区的第一行进行扫描,一旦发现这行中包含欲查询的字符串,那么该文件名以及注释的第一行将被显示。当然,用户想建立自己文件的在线帮助时,也可利用这种机制。

2.5.3 其他帮助指令

MATLAB 还提供了其他一些帮助指令,见表 2 5-1。

表 2.5-1 其他帮助指令

exist	检查指定名字的变量或函数文件的存在性。
what	按扩展名分类列出(在搜索路径中)指定目录上的文件名。
which	列出指定名字文件所在的目录。
who	列出工作内存的变量名(详见 2 4 6 节)。
whos	列出工作内存的变量名及其细节(详见 2 4 6 节)。

关于这些指令详细的情况在此不多介绍,有兴趣的读者可以用 help 指令自行查阅。

2.6 用户目录的建立和搜索路径

本节将回答两个问题:如何建立用户自己的工作目录? 如何与那些不在原先设定路径上的目录交换信息?

2.6.1 用户工作目录的建立

为了保护 MATLAB 目录结构的严整,为了管理用户自己用 MATLAB 所创建、修改的 M 文件和其他文件的方便,用户应该建立自己的工作目录。

由于 MATLAB 启动后的缺省目录是 C:\MATLAB\BIN,所以若不建立自己的工作目录,那么在 MATLAB 环境下所产生的数据文件就很可能登录在这缺省目录上。这是很不好的,应尽量避免。

与一般目录创建方法相同，用户工作目录有两种创建方法。以建立工作目录 C：\ MY-DIR 为例说明如下：

(1) DOS 环境下的创建方法

在 DOS 环境下，键入以下命令：

>md mydir

(2) Windows 环境下的创建方法

打开【主群组】中的【文件管理器】；用【文件管理器】中文件(File)栏下拉菜单中的创建目录(Create Directory)选项建立目录 C：\ mydir 。

2.6.2 搜索路径的扩展

假如不专门设定，那么 MATLAB 只可能在它启动时(由 matlabrc m)设定的路径上搜索(见2 3 2节)。在这种情况下，用户无法与原定路径以外的其他目录交换信息。即便用户已经建立了自己的工作目录，此目录仍然无法与 MATLAB 交换信息。鉴于此，搜索路径的扩充是必需的。下面介绍三种扩充方法。

将用户目录指定为当前工作目录

在 MATLAB 指令窗里，输入以下指令：

cd c：\ mydir

说明：

(1) 该指令执行后，用户目录 c：\ mydir 就成了当前工作目录。

(2) 此后，在 MATLAB 指令窗里，可以调用 c：\ mydir 目录上的所有文件。

(3) 在 MATLAB 指令窗里生成的数据文件也将缺省地登录在该目录上。

(4) MATLAB 4 0 及更高版本都有这个指令。

利用 path 指令扩展搜索路径

若想把用户目录 c：\ mydir 纳入搜索路径，那么可用下述指令实现：

path(path, 'c· \ mydir')

说明：

(1) 上述指令执行后，在 MATLAB 指令窗里就可以调用目录 c：\ mydir 上的文件。但请记住：此时 c：\ mydir 并不是当前工作目录。

(2) path 指令还有其他功用，详细请看 MATLAB 用户指南或在线求助。

(3) 用 path 指令扩展的搜索路径仅在当前 MATLAB 工作环境下有效。这也就是：若用户退出当前 MATLAB 后，再重新启动 MATLAB，那么在前一环境下用 path 所定义的扩展搜索路径无效。

(4) MATLAB 4 0 及更高版本都有 path 指令。

利用 pathtool 修改搜索路径

MATLAB 4 2 版提供了一个永久性修改 MATLAB 搜索路径的专用工具 pathtool 。若用

户想把自己工作目录 c:\mydir 永久性地设置在 MATLAB 搜索路径上，那么可按以下步骤进行：

（1）在 MATLAB 环境下，键入指令

pathtool

（2）该指令执行后，出现如图 2.6-1 所示的窗口。

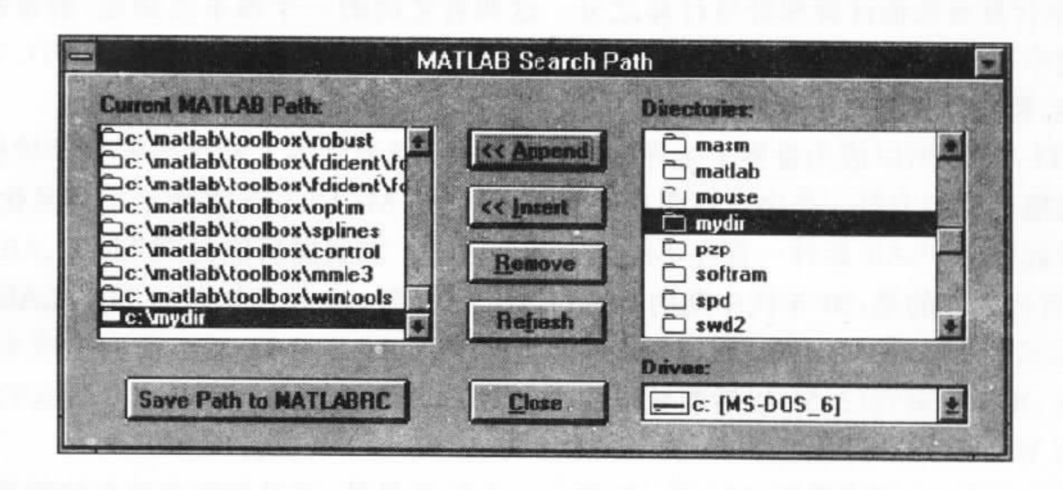


图 2.6-1　修改搜索路径的工具窗

（3）在驱动器(Drives)栏和目录(Directories)栏中选定用户目录 c:\mydir。

（4）用鼠标点击加接(Append)键，在左侧的当前路径(Current MATLAB path)栏的最下方便出现 c:\mydir。

（5）用鼠标点击保存路径(Save path to MATLABRC)键。

（6）用鼠标点击关闭(Close)键，就完成了搜索路径的扩展。

关于 pathtool 的其他使用方法，请见第六章。

第三章 MATLAB 的数值计算功能

数学计算有数值计算和符号计算之分。这两者之间的一个根本区别是：前者的表达式、矩阵变量中不允许包含未定义的自由变量，而后者允许。本章将专门介绍 MATLAB 的数值计算功能，符号计算则是下章的内容。

MATLAB 之所以成为世界上主导数值计算软件，成为世界上最优秀的数学软件之一，其出色的数值计算能力是一个决定性因素。自 80 年代初 Mathworks 公司推出商品化MATLAB 数学软件起，MATLAB 软件一直在不断地完善和改进。在我国所能见到 MATLAB 的各种版本中，较有代表性的是：80 年代中期的 MATLAB 3 0x 版；90 年代初期的 MATLAB 3 5x 版；1993 年以后的 MATLAB 4 0x 版；1994 年问世的 MATLAB 4 2x 版。前两个版本中的绝大部分都在 DOS 下运行(3 5k 以后版勉强能在 WINDOWS 上工作)，而从 4 0 版起的MATLAB 都运行在 WINDOWS 平台上。

纵观这些版本，不难看到：MATLAB 作为一个软件品种，其早期版本就有的数值计算核心技术不但至今仍在发挥作用，而且在不断地改进下更加充满活力。本章将以 4 2 版为蓝本介绍 MATLAB 的数值计算功能；又考虑软件的向下兼容，适度地指出不同版本之间的差别。

在具体介绍本章内容之前，再次提醒读者注意以下两点：

(1) MATLAB 是以矩阵(或称作数组)为运算“单元”。这给编写程序带来很大方便。在数值计算中，运算“单元”定义在复数域上，因此在数值计算中尤能强烈地感受到 MATLAB 这种设计思想带来的种种好处。

(2) 在 MATLAB 中，不管是数值矩阵还是符号矩阵都不必事先定义维数大小。MATLAB 会根据用户所输入的矩阵结构自动配置，并在此后的运算中按正确的数学法则自动地调整矩阵的维数。

由于本章内容是在 MATLAB 的活笔记本(MATLAB Notebook)中编写的，因此所有例题中的指令都是实际可运作的，其结果都是实际运算得出的。假如读者在阅读本书的同时能自己动手做这些例题，那么定能对 MATLAB 的功能有更深切的体会。

3.1 数值矩阵的创建、保存和数据格式

在 MATLAB 中，我们把由下标表示次序的标量数的集合称为矩阵(Matrix)，或称为数组(Array)。从孤立的数的集合角度看，不管是称矩阵还是称数组，它们所指的没有什么不同。但从运算角度看，矩阵运算和数组运算则是两类不同的运算。

本节系统地介绍矩阵(或称数组)的不同创建方法。为简洁起见，就不再反复同时列出矩阵、数组的名字，而统称矩阵。本节的前五小节介绍矩阵的各种创建方法。其中，除第 3.1.2 节介绍的矩阵编辑器法为 4 2 版所特有外，其余各小节内容对各个版本 MATLAB 都适用。第六小节介绍输出数据格式。

3.1.1 创建数值矩阵的直接输入法

对于较小较简单的矩阵，从键盘上直接输入矩阵是最常用、最方便和最好的数值矩阵创建方法。用这种直接法送入的矩阵由以下三个要素组成：

(1) 整个输入矩阵必须以方括号“[ ]”为其首尾；

(2) 矩阵的行与行之间必须用分号“;”或回车键【Enter】隔离；

(3) 矩阵元素必须由逗号“,”或空格分隔。矩阵元素可以是不包含未定义变量的任何表达式。但在 MATLAB 中，即便没有任何元素的矩阵也是合法的，它被称为“空阵(Empty matrix)”。空阵在 MATLAB 的矩阵操作中非常有用(详见第 3.6.3 节)。

【例 1】在 MATLAB 环境下，用下面三条指令创建数值矩阵 C。

```
a=2.7358; b=33/79;            % 这两条指令分别给变量 a，b 赋值
C=[1,2*a+i*b,b*sqrt(a);sin(pi/4),a+5*b,3.5+i]
                              % 这指令用于创建矩阵 C
```

(下面是屏幕的显示结果)

```
C =
    1.0000     5.4716 + 0.4177i   0.6909
    0.7071     4.8244             3.5000 + 1.0000i
```

说明：

(1) 分号“;”的三个作用：

(A) 在“[ ]”方括号内时，它是矩阵行间的分隔符。

(B) 它可用作在指令与指令间的分隔符。

(C) 当它存在于赋值指令后时，该指令执行后的赋值结果将不显示在屏幕上。

(2) 指令中的“pi”和 i 都是 MATLAB 的预定义变量(Predefined variable)名。

(A)“pi”代表 π 的一个拥有 16 位有效数字的近似值。

(B)“i”代表定义为 $i=\sqrt{-1}$ 的虚数单位。

(3) 在 MATLAB 中，“%”以后的物理行中的内容仅作注释用，对 MATLAB 的计算不产生任何影响。

(4) 被赋过值的变量，不管是否在屏幕上显示过，都存放在 MATLAB 的工作内存(workspace)中，可随时被以后的指令所调用或显示。

(5) 当然，在每个指令输入行结束后，要按回车键，才能使该行指令被 MATLAB 执行。由于这是使用计算机最起码的常识，所以在此后，本书将不再重申这一必需的操作步骤。

【例 2】复数矩阵的另一种输入方式。

```
R=[1,2,3;4,5,6],I=[11,12,13;14,15,16]
CN=R+i*I
R =
     1     2     3
```

```
     4      5      6
I =
    11     12     13
    14     15     16
CN =
   1.0000 +11.0000i  2.0000 +12.0000i  3.0000 +13.0000i
   4.0000 +14.0000i  5.0000 +15.0000i  6.0000 +16.0000i
```

说明：

(1) 逗号“，”的三个作用：

(A) 在“[ ]”方括号中时，它可作为元素与元素间的分隔符。

(B) 它可用作在指令与指令间的分隔符。

(C) 当它存在于赋值指令后时，该指令执行后的赋值结果将显示在屏幕上。

(2) 本例的“i”和“I”表明：在 MATLAB 中，同一个字母的大小写确实具有不同的含义。

3.1.2　利用矩阵编辑器创建和修改数值矩阵

当需输入矩阵的维数较大或者矩阵元素不是很简单的数字和已赋值变量时，若用直接输入法创建数值矩阵就会感到烦琐不便，并可能产生一些较难发现的误操作。为改善用户工作环境，MATLAB 4.2 提供了一个专门的数值矩阵编辑器（Matrix editor）。当该矩阵编辑器被调用时，将即时弹出一个如图 3.1-1 所示的 MATLAB 矩阵（MATLAB Matrix）视窗。该数值矩阵编辑器的调用指令和格式如下：

edit Vname　　　数值矩阵编辑器调用指令（Vname 是新或已定义的变量名）

【例 1】演示输入一个维数为(3×6)的矩阵 B 的中间过程。

```
edit B                   % 调用编辑器给 B 变量赋一个数值矩阵的指令
```

当输入完以上指令并按回车键(Enter)以后，就出现一个矩阵 B 视窗。图 3.1-1 是向 B 阵输入第(1,4)元素但尚未确认时的状态。

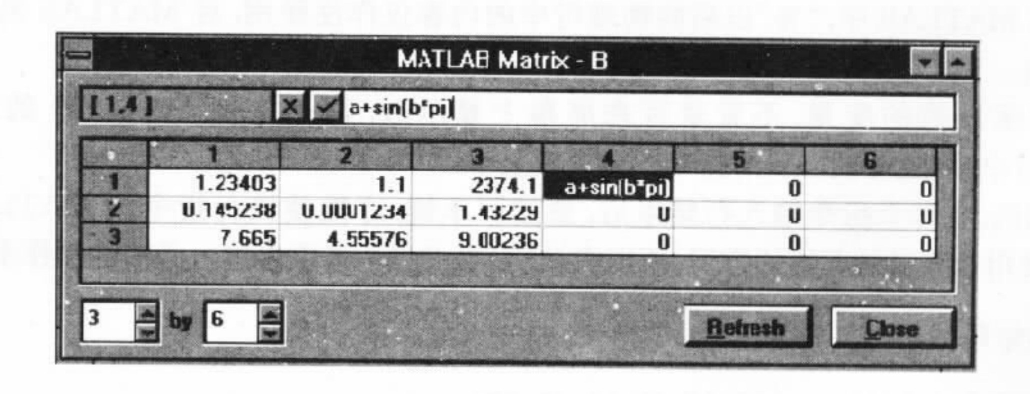


图 3.1-1　数值矩阵编辑器

下面是对此视窗的注释：

(1) 该视窗标题行(Title bar)中最后一个字母“B”是待定义的变量名。

(2) 在该窗左下角的两个滚动盒(Scroll box)中的“3”和“6”分别是B阵的行维和列维。对于尚未定义的变量来说，该视窗刚出现时的缺省维数是(5×5)。这维数值可根据用户要求调整。

(3) 该窗上方第二行最左边的“[1,4]”是正待输入的元素标号(Index)，而右边的白色部分是正在输入的表达式。该第二行中间的两个按键中，那打蓝色勾的是“确认输入键”，而打红色叉的是“取消键”。当然在输入表达式中出现的变量名必须是已被赋值的。

(4) 视窗正中的白色长方分格块是B阵元素显示区。尚未送入数值的元素被缺省地定义为零。而其中那呈蓝色的长方小格是正待输入的元素。一旦该元素的输入内容被认定正确，按下“确认输入键”或按键盘上的回车键，此后该元素位置上出现的将总是表达式的计算结果，而不是表达式本身。

(5) 矩阵元件的输入次序可由键盘上的“方向移动键”或鼠标选定。在所选定的元素长方小格四周会出现虚线框。

(6) 右下角的“Refresh”键是用来调节矩阵元素显示区的大小的。

(7) 当所有矩阵元素被完全确认后，按下“Close”键，于是B阵就被定义保存，以备后用。

当然，矩阵编辑器也可用来对已定义矩阵的元素加以修改。还可以把原定义矩阵“裁剪”为它的左上主子阵，或反过来把原阵“扩展”成以它为左上主子阵的更大矩阵。

3.1.3 利用MATLAB函数和语句创建数值矩阵

从广义上讲，整个MATLAB软件就是为用户能便捷得到数值矩阵结果而服务的；从狭义上看，MATLAB确实提供了许多生成和操作矩阵的函数。用户可以利用它们去创建数值矩阵，并且随用户对MATLAB不断熟悉，利用MATLAB现有函数和语句创建矩阵的方法就会愈多。在本小节中，只是举几个算例来说明这种矩阵创建方法，而不对指令本身作过于详细的介绍。

【例1】利用指令 reshape 创建数值矩阵。

```
av=1:12                %产生有12个元素的行向量av
bm=reshape(av,3,4)  %利用向量av创建(3×4)矩阵bm
av =
     1     2     3     4     5     6     7     8     9    10    11    12
bm =
     1     4     7    10
     2     5     8    11
     3     6     9    12
```

【例2】利用指令 diag 产生对角阵。

```
ar=rand(3,3)          %产生(3×3)的“0-1均匀分布”随机阵ar
d=diag(ar)            %用阵的主对角元形成向量d
D=diag(d)             %用向量元素构成对角阵D
```

```
ar =
    0.2190    0.6793    0.5194
    0.0470    0.9347    0.8310
    0.6789    0.3835    0.0346
d =
    0.2190
    0.9347
    0.0346
D =
    0.2190         0         0
         0    0.9347         0
         0         0    0.0346
```

【例 3】由 4 元行向量[1 1 79 2 17 1]创建一个伴随阵。

```
p=[1, 1.79, 2.17, 1];                  %送入 4 元向量 p
cp=[zeros(3,1) eye(3,3);p]             %创建伴随阵 cp
cp =
         0    1.0000         0         0
         0         0    1.0000         0
         0         0         0    1.0000
    1.0000    1.7900    2.1700    1.0000
```

说明：在上述指令中的 zeros(3,1) 是生成三元全零列向量指令；eye(3,3) 是生成三维单位阵的指令。关于它们的详细说明，请查 MATLAB 用户手册，也可在 MATLAB 环境中用 help zeros 及 help eye 获得在线帮助说明。

3.1.4 利用 M 文件创建和保存矩阵

本小节所介绍的方法，既适用于数值矩阵，又适用于符号矩阵。

对于今后经常需要调用的矩阵，尤其是比较大且比较复杂的矩阵，为它专门建立一个 M 文件是值得的。下面通过一个简单例子来说明这种 M 文件的创建过程。

【例 1】创建和保存矩阵 AM 的 MyMatrix m 文件生成过程。

步骤一：使用 Dos 编辑器(edit)、Windows 书写器(write)、记事本(notepad)或其他字处理软件(如 Word 等)下敲入以下内容。

```
% MyMatrix m      Creation and preservation of matrix AM
AM=[101,102,103,104,105,106,107,108,109;201,202,203,204,205,206,207,…
    208,209;301,302,303,304,305,306,307,308,309];
```

步骤二：把此内容以纯文本方式(ASCII 码)保存在用户自己目录下名为 MyMatrix m 的文件中。

步骤三：在 MATLAB 指令窗中，只要敲人 MyMatrıx ，矩阵 AM 就会自动生成于 MATLAB 工作内存中(即产生一个名为 AM 的变量)，供以后显示和调用。

说明：

(1) MyMatrıx m 文件中以符号“%”开头的第一行是注释行。注释行的格式：通常，总在“%”符号后紧接着写该 M 文件的名字(但并不是必须的)；在这之后，留一定空间，再写该文件的主要功用。注释文字不限中文和西文。

MATLAB 的所有 M 文件一般都是这种结构。这种注释为 MATLAB 的在线求助指令 help 和 lookfor 提供信息源(详见第八章)，并可增加 M 文件的可读性。

(2) 当一个指令长度超出物理行时或当为增加可读性而需把一个指令分行书写时，可用由三个以上小黑点组成的续行号“ ”实现。

(3) 该例仅为示意，故把 AM 阵取得十分简单。事实上，这种方法对创建和保存的矩阵的大小没有限制。

3.1.5 通过 MAT 文件保存和获取矩阵

有时需要把当前 MATLAB 工作内存中的一些有用数据长久地保留下来。在这种场合，MAT 文件就可发挥作用。MAT 文件是 MATLAB 保存数据的一种标准格式二进制文件。MAT 文件的生成和调用由指令 save 和 load 进行。

【例 1】借助 mydata mat 文件保存第 3 1 1 小节两个算例中所产生的矩阵 A 和 CN。

步骤一：在矩阵 A、CN 存在于 MATLAB 内存空间的前提下，敲人以下指令便可达到目的：

```
save mydata A CN
```

步骤二：假如在下次重新进人 MATLAB 后，需要使用 A、CN 矩阵，那么应先用下述指令把 mydata mat 中的内容读人 MATLAB 工作内存·

```
load mydata
```

步骤三：在上述指令执行后，在当前的 MATLAB 环境中，A 和 CN 就是两个已知变量了。

说明.

(1) mydata 是由用户自已给的文件名，MATLAB 默认扩展名为 mat。上述 save 指令执行以后，该 mydata mat 文件将被登录在 matlab \ bın 子目录上。假如用户有意要让 mydata mat 登录在指定的目录(比如 D 盘的 MYSUB 目录)上，那么必须把步骤一中的指令改写成

```
save d:\mysub\mydata A CN
```

当然，以后的 load 指令后的内容也要作相应的改变。

(2) 本节所介绍的方法也适用于符号矩阵。

(3) 事实上，数值矩阵还能借助 save 指令以 ASCII 码形式保存在文件中，以供 MATLAB 或其他外部程序使用；同样也可借助 load 指令把(不管是由 MATLAB 生成的还是其他外部程序生成的)以 ASCII 码形式保存的数据读取出来。详见第八章。

3.1.6 数据输出格式

在 MATLAB 中，数据的存储和运算都是以双精度进行的，而屏幕上显示的运算结果却可以用不同的格式表示。控制数据显示格式的指令是 format，具体格式类型和运用方法见表 3 1-1。

表 3.1-1 format 指令的格式类型

指令格式	显示结果	举例说明
format format short	5 位定点表示	3 1416
format long	15 位定点表示	3 14159265358979
format short e	5 位浮点表示	3 1416e+00
format long e	15 位浮点表示	3 14159265358979e+00
format rat	近似有理数表示	355/113
format hex	十六进制表示	400921fb54442d18
format +	表示大矩阵用，+ 、- 、空格分别表示正数、负数、零元素	+
format bank	(金融)元、角、分表示	3 14
format compact	在显示变量之间没有空行，这样可以节省显示屏幕	
format loose	在显示变量之间有空行	

说明：假如矩阵元素都是整数，那么在上述前五种格式下，都保持整数显示不变。

3.2 矩阵运算和数组运算

矩阵运算和数组运算是 MATLAB 的数值运算中的两大类运算。矩阵运算是指按矩阵运算法则进行的运算；数组运算则无论是哪种运算操作都是对元素逐个进行的。

MATLAB 设计这两种运算的目的：(1)使 MATLAB 的运算指令的形式尽可能地与标准教科书的数学表达一致，便于用户掌握；(2)使大批数据处理的表现与标量情况相似，大大简化编程，并便于阅读；(3)提高数据处理的效率和可靠性。

3.2.1 矩阵运算和数组运算指令对照汇总

本节集中、扼要地罗列两种运算指令形式和实质的异同点。

表 3.2-1 两种运算指令形式和实质内涵的异同表

矩阵运算指令	指 令 含 义	数组运算指令	指 令 含 义
A'	矩阵共轭转置	A.'	矩阵元素的共轭转置，相当于 conj(A')

表 3.2-1(续)

矩阵运算指令	指 令 含 义	数组运算指令	指 令 含 义
A+B	矩阵相加	**A.+B**	对应元素相加(与矩阵相加等效)
A−B	矩阵相减	**A.−B**	对应元素相减(与矩阵相减等效)
s+B	标量和矩阵相加(MATLAB 约定的特殊运算)		
s−B, B−s	标量和矩阵相减(MATLAB 约定的特殊运算)		
A*B	内维相同矩阵的乘	**A.*B**	同维数组对应元素相乘
s*A	A 的每个元素乘 s	**s.*A**	A 的每个元素乘 s(标量和矩阵相乘结果与此同)
A/B	A 右除 B	**A./B**	A 的元素被 B 的对应元素除
B\A	A 左除 B	**B.\A**	与上相同
inv(B)	B 阵的逆		
		s./B, B.\s	s 分别被 B 的元素除
A^n	A 阵的 n 次幂	**A.^n**	A 的每个元素自乘 n 次
A^p	A 阵的非整数乘方(据矩阵特征值分解定义)	**A.^p**	A 的每个元素分别求非整数乘方
p^A	标量的矩阵乘方(据矩阵特征值分解定义)	**p.^A**	以 p 为底,分别以 A 的元素为指数,求幂
expm(A)	A 的矩阵指数函数(据矩阵特征值分解定义)	**exp(A)**	以自然数 e 为底,分别以 A 的元素为指数,求幂
logm(A)	A 的矩阵指数函数(据矩阵特征值分解定义)	**log(A)**	对 A 的各元素求对数
sqrtm(A)	A 的矩阵指数函数(据矩阵特征值分解定义)	**sqrt(A)**	对 A 的各元素求平方根
funm(A,'FN')	一般矩阵函数。在此 FN 是合法的函数名,据矩阵特征值分解定义	**f(A)**	求 A 各个元素的函数值。在此 f() 表示各种基本函数
		A#B	A、B 阵对应元素间的关系运算。在此,# 代表各种关系运算符
		A@B	A、B 阵对应元素间的逻辑运算。在此,@ 代表各种逻辑运算符

说明:

(1) 数组四则运算、乘方、转置运算符中的小黑点绝对不能遗漏,否则将不按数组运算规律进行计算。

(2) 不管执行什么数组运算,所得计算结果数组总是与参与运算的数组同维。

(3) 要特别注意两种运算在以下方面的本质区别:乘、除、乘方、三角和超越函数。详见 3 2 3和3 3 节 。

(4) 关系运算和逻辑运算仅对数组进行。详见第 3.4 节。

(5) 由于加、减、转置等运算非常简单,指令也容易理解,所以本书不给算例。下面给出两

个示例，用户可以从中体会到 MATLAB 的一些特点。

【例 1】矩阵运算示例：计算 CA^2B。

```
rand('seed',0)          % 为保证算例可重复性，设置随机数的种子为 0
A=rand(5,5);            % 产生(5×5)随机阵
B=rand(5,3);            % 产生(5×3)随机阵
C=rand(2,5);            % 产生(2×5)随机阵
M=C*A^2*B               % 计算 CA^2B
M =
    5.9669    7.6491    7.7459
    6.4265    8.2881    8.3650
```

【例 2】数组运算在函数 $y=e^{-t}\sin t$ 可视化中的应用(图 3.2-1)。

```
t=0:pi/40:4*pi;             % 产生维数为(1×161)的数组变量 t
y=exp(-t).*sin(t);          % 运用数组运算计算(1×161)维的数组变量 y
plot(t,y)                   % 在直角坐标上画 y=e^-t sint 曲线
```

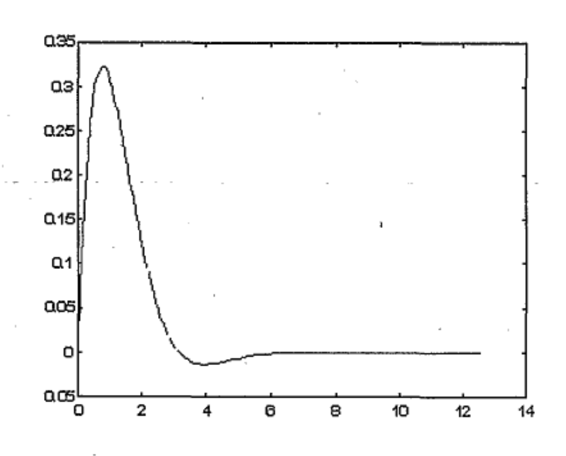


图 3.2-1 $y=e^{-t}\sin t$ 曲线

3.2.2 矩阵乘和数组乘

两种乘法运算的异同如下：

(1) 矩阵与矩阵相乘

相乘矩阵的维数必须相等，即满足运算规则 $C_{m\times n}=A_{m\times k}B_{k\times n}$。

(2) 数组与数组相乘

相乘数组必须有相同的行维和列维。数组 $A_{m\times n}$ 与 $B_{m\times n}$ 相乘是指这两个数组的相应元素

相乘,即数组相乘的结果为 $C_{m\times n}=\begin{bmatrix} a_{11}\times b_{11} & \cdots & a_{1n}\times b_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1}\times b_{m1} & \cdots & a_{mn}\times b_{mn} \end{bmatrix}$。

(3)"标量与矩阵相乘"的结果和"标量与数组相乘"的结果相同。

【例 1】矩阵乘和数组乘的不同。

```
A=[11 12 13 14 15;16 17 18 19 20];
B=[1 2 3 4 5;6 7 8 9 10];
C=A*B'
D=A .*B
C =
    205     530
    280     730
D =
    11     24     39     56     75
    96    119    144    171    200
```

3.3 矩阵除和数组除

在线性代数教程中,有关于矩阵逆的定义,但没有矩阵除的运算。而在 MATLAB 中,矩阵除运算却有十分丰富的内涵。根据实际问题的需要,MATLAB 提供两种除法指令:左除和右除。由于 MATLAB 是根据关系式:B \ A=(A'/B')' 来定义和设计右除的,所以下面的讨论仅针对左除展开。

对于方程 Ax=b,A 是(n×m)的矩阵,那么

(1) 当 n=m 且非奇异时,此方程称为"恰定"方程(Properly Determaned Equation);

(2) 当 n>m 时,此方程称"超定"方程组(Overdetermaned Equation);

(3) 当 n<m 时,此方程称"欠定"方程组(Underdetermaned Equation)。

矩阵除可以很容易地、很有效地解上述三种方程。有关内容见 3.3 1 ,3 3.2 ,3.3.3 节。第 3 3 4 节将介绍数组除。

3.3.1 矩阵逆和用除法解恰定方程组

方阵的逆

为说明除运算的意义,先介绍求逆指令。在 MATLAB 中,对于任何方阵都可用下述指令求逆。

inv(A)　　求矩阵 A 的逆;A 必须是方阵。

说明:

(1) 由于 MATLAB 遵循 IEEE 算法，所以即使 A 阵奇异，该运算也照样进行。但在运行结束时，一方面给出警告："Warning: Matrix is singular to working precision"；另一方面，所得逆阵的元素将都是"Inf"(无穷大)。

(2) 当 A 阵"病态"时，也会给出警告信息。

(3) 在 MATLAB 中，inv 指令不很有用。MATLAB 推荐：尽量使用除运算，少用逆运算。理由在下面几节介绍。

用除法运算解恰定方程组

现采用举例方式，讨论解恰定方程组的下述两种方法。

(1) 采用求逆运算解方程。其指令为

x = inv(A) * b

(2) 采用左除运算解方程

x = A \ b

【例 1】"求逆"法和"左除"法解恰定方程的性能对比。

为对比这两种方法的性能，先用以下指令构造一个条件数很大的高阶恰定方程。

```
rand('seed',12);             % 选定随机种子，目的是可重复产生随机阵 A
A= rand(100)+1.e8;           % rand(100)生成(100×100)均匀分布随机矩阵
                             % 每个随机阵元素加一个数的目的是使 A 阵条件数升高
x=ones(100,1);               % 令解向量 x 为全 1 的 100 元列向量
b=A*x;                       % 为使 Ax=b 方程一致，用 A 和 x 生成 b 向量
cond(A)                      % 求 A 阵的条件数
ans =
   5.0482e+011
```

计算表明 A 阵的条件数约为 5×10^{11}。

过程一："求逆"法解恰定方程的误差、残差和所用时间。

```
tic                          % 启动计时器(Stopwatch Timer)
xi=inv(A)*b;                 % xi 是用"求逆"法解恰定方程所得的解
toc                          % 关闭计时器，并显示解方程所用的时间
eri=norm(x-xi)               % 解向量 xi 与真解向量 x 的范-2 误差
rei=norm(A*xi-b)/norm(b)
                             % 方程的范-2 相对残差
elapsed_time =
      0.8700
eri =
   1.0044e-004
rei =
   1.2177e-006
```

过程二："左除"法解恰定方程的误差、残差和所用时间。

```
tic
xd=A \ b;                    % xd 是用“左除”法解恰定方程所得的解
toc
erd=norm(x-xd)
red=norm(A * xd-b)/norm(b)
elapsed_time =
     0.3300
erd =
   2.7502e-005
red =
   5.7220e-016
```

说明：

(1) 计算结果表明：除法求解方程的速度是逆阵法解方程速度的 2 5 倍，所得解的精度相当；但“除法”的相对残差几乎是“机器零”，而“逆阵法”的相对残差高得多。

(2) MATLAB 在设计求逆函数 inv 时，采用的是高斯消去法(Gaussian elimination)。

(3) MATLAB 在设计除运算解恰定方程时，并不求逆，而是直接采用高斯消去法求解，有效地减少了残差，所以即便在高条件数下也能得到较好的结果。

3.3.2 用除法运算解超定方程

解超定方程也有两种途径：

(1) 求正则方程(Normal equations)$(A^TA)x=A^Tb$ 的解。

(2) 用 Householder 变换(Householder Transformation)直接求原超定方程的最小二乘解。

由于第二种方法采用的是正交变换，据最小二乘理论可知，第二种方法所得解的准确性、可靠性都比第一种方法好得多。MATLAB 解超定方程的除运算法用的就是第二种方法。下面举例说明其使用方法。

【例 1】除运算解超定方程的简单算例。

```
A=[1 2 3;4 5 -6;7 8 9;10 11 12];b=(1:4)';
x=A \ b
x =
  -0.3333
   0.6667
   0.0000
```

3.3.3 用除法运算解欠定方程

欠定方程的解是不唯一的。用除法运算所得的解将有两个重要特征：在解中至多有 Rank (A) 个非零元素；它是这种类型解中范数最小的一个。

【例 1】解欠定方程。

```
B=A';c=[1 3 3]';          % 这 A 阵取自上小节例 1
x=B\c
x =
  2.0000
  0.1667
       0
-0.1667
```

说明：

(1) 所得结果表明：此解实际上是向量 c 在 B 阵第 1，2，4 列所张空间中的投影。它的范数长度为 2.0138 。

(2) 分别计算向量 c 在 B 阵其他三个由三个列向量所张空间中的投影的范数长度，它们分别是 2 1049，7 3068，8 4255e+015 。它们都比前者大。

3.3.4 数组除

数组除有以下一些使用格式：

A./B，B \ A 执行运算 $\begin{bmatrix} a_{11}/b_{11} & \cdots & a_{1n}/b_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1}/b_{m1} & \cdots & a_{mn}/b_{mn} \end{bmatrix}$

A./s 执行运算 $\begin{bmatrix} a_{11}/s & \cdots & a_{1n}/s \\ \vdots & \vdots & \vdots \\ a_{m1}/s & \cdots & a_{mn}/s \end{bmatrix}$

s./B，B \ s 执行运算 $\begin{bmatrix} s/b_{11} & \cdots & s/b_{1n} \\ \vdots & \vdots & \vdots \\ s/b_{m1} & \cdots & s/b_{mn} \end{bmatrix}$

3.4 矩阵乘方和数组乘方

3.4.1 方阵的标量乘方和数组的标量乘方

方阵的标量乘方 A^p

情况一：当 p 取整数时，该指令的运算结果可作如下理解：

(1) 当 p>0 时，A^p 指令可直接理解为方阵 A 自乘 p 次的结果。

(2) 当 p<0 时，A^p 指令可理解为方阵 A 自乘 p 次后的逆。当然只有非奇异阵有这种运算。

(3) 当 p=0 时，A^0 将给出与 A 阵同维的单位阵。

情况二：当 p 为非整数时，它遵循以下规则：

若有特性值分解 AV=VD，那么定义 $A^p = V\begin{bmatrix} d_{11}^p & & \\ & \ddots & \\ & & d_{nn}^p \end{bmatrix}V^{-1}$。其中，D 为对角阵。

说明：

(1) 对于有重根的阵，以上指令不适用。

(2) 指令 A^p 是据特征根分解原理设计的，而 sqrtm(A) 是据 Schur 分解设计的。

(3) 从理论上讲，对于有些矩阵的非整数次方，可能存在多个解，而 MATLAB 指令仅给出其中一个解。以 A^0 5 和 sqrtm(A) 为例，它们仅取 A 阵所有特征根的正开平方根为解。

【例 1】求矩阵的非整数乘方，并验证。

```
A=[1 2 3;4 5 6;7 8 9];
Ap=A^0.3
Ap =
     0.6962 + 0.6032i    0.4358 + 0.1636i    0.1755 - 0.2759i
     0.6325 + 0.0666i    0.7309 + 0.0181i    0.8292 - 0.0305i
     0.5688 - 0.4700i    1.0259 - 0.1275i    1.4830 + 0.2150i
```

下面是按矩阵非整数乘方的定义进行验证的指令：

```
[V,D]=eig(A);
AAp=V*diag([D(1,1)^0.3,D(2,2)^0.3,D(3,3)^0.3])/V,
AAp =
     0.6962 + 0.6032i    0.4358 + 0.1636i    0.1755 - 0.2759i
     0.6325 + 0.0666i    0.7309 + 0.0181i    0.8292 - 0.0305i
     0.5688 - 0.4700i    1.0259 - 0.1275i    1.4830 + 0.2150i
```

说明：当一个实方阵有复数特征值或负实特征值时，该方阵的非整数次方就可能是复数阵。

<u>数组的标量乘方 A.^p</u>

(m×n)数组 A 的标量 p 乘方，定义为 $\begin{bmatrix} a_{11}^p & \cdots & a_{1n}^p \\ \vdots & \vdots & \vdots \\ a_{m1}^p & \cdots & a_{mn}^p \end{bmatrix}$。

【例 2】数组的标量乘方。

```
Aap=A.^0.3
Aap =
    1.0000    1.2311    1.3904
    1.5157    1.6207    1.7118
    1.7928    1.8661    1.9332
```

说明：对比例 1 和例 2 的计算结果，可见两种运算是截然不同的。

3.4.2 标量的矩阵乘方和标量的数组乘方

标量的矩阵乘方 p^A

求标量 p 的矩阵乘方定义为 $p^A = V \begin{bmatrix} a^{d_{11}} & & \\ & \ddots & \\ & & a^{d}_{m} \end{bmatrix} V^{-1}$，式中 V，D 取自特征值分解 AV = AD。

【例 1】求标量的矩阵乘方。

```
pA=(0 3)^A                % A 阵取自上小节的算例
pA =
     2.9342     0.4175    -1.0993
    -0.0278     0.7495    -0.4731
    -1.9898    -0.9184     1.1531
```

标量的数组乘方 p.^A

标量的(m×n)数组乘方定义为 $\begin{bmatrix} p^{a11} & \cdots & p^{a1n} \\ \vdots & & \vdots \\ p^{am1} & \cdots & p^{amn} \end{bmatrix}$。

【例 2】标量的数组乘方。

```
paA=0.3.^A
paA =
    0.3000    0.0900    0.0270
    0.0081    0.0024    0.0007
    0.0002    0.0001    0.0000
```

【例 3】在新旧版本中，数组运算书写方式的对比。

```
Z=2 ^[1 2 3 4 5 6]
Z =
     2     4     8    16    32    64
```

说明：

(1) 该算例是 MATLAB 各版本用户指南中都有的。

(2) 在 3 0 版和 3 5 版中，特别强调书写计算指令时，在“2”之后，“ ^”之前必须有空格，否则或无法运算，或不能得到正确的运算结果。

(3) 在 MATLAB 4 0 以上版本里，“ ^”之前有无空格都能正确运算。

3.5 数组函数和矩阵函数

在 MATLAB 所提供的函数有两类：一类按数组运算法则设计的，称之为数组函数，表示为 f(·)；另一类按矩阵运算法则设计的，称为矩阵函数，表示为 funm(·)。

3.5.1 基本数组函数

数组函数是按以下规则设计的：

对于 $A=\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}$，定义 $f(A)=\begin{bmatrix} f(a_{11}) & \cdots & f(a_{1n}) \\ \vdots & \vdots & \vdots \\ f(a_{m1}) & \cdots & f(a_{mn}) \end{bmatrix}$。

满足这个定义的基本函数指令罗列于表 3 5-1 和表 3 5-2 中。

表 3.5-1 f(·)基本函数表

函数名称	功 能	函数名称	功 能
sin	正弦	acosh	反双曲余弦
cos	余弦	atanh	反双曲正切
tan	正切	acoth	反双曲余切
cot	余切	asech	反双曲正割
sec	正割	acsch	反双曲余割
csc	余割	fix	朝零方向取整
asin	反正弦	ceil	朝正无穷大方向取整
acos	反余弦	floor	朝负无穷大方向取整
atan	反正切	round	四舍五入到整数
atan2	四象限反正切	rem	除后取余数
acot	反余切	sign	符号函数
asec	反正割	abs	绝对值
acsc	反余割	angle	复数相角
sinh	双曲正弦	imag	复数虚部
cosh	双曲余弦	real	复数实部
tanh	双曲正切	conj	复数共轭
coth	双曲余切	log10	常用对数
sech	双曲正割	log	自然对数
csch	双曲余割	exp	指数
asinh	反双曲正弦	sqrt	平方根

表 3.5-2 f(·)特殊函数表

函数名称	功 能	函数名称	功 能
bessel	第一、第二类 Bessel 函数	erf	误差函数
beta	Beta 函数	eerfinv	逆误差函数
gamma	Gamma 函数	ellipk	第一、第二类全椭圆积分
rat	有理近似	ellipj	Jacobi 椭圆函数

3.5.2 基本矩阵函数

MATLAB 所提供的基本矩阵函数指令见表 3.5-3。

表 3.5-3 基本矩阵函数指令

函数指令	指令含义	函数指令	指令含义
cond(A)	A 阵的条件数(A 的最大奇异值除以最小奇异值)	svd(A)	A 阵的奇异值分解
det(A)	方阵 A 的行列式值	trace(A)	矩阵 A 的迹
dot(A, B)	矩阵的点积	expm(A)	矩阵指数 e^A
eig(A)	矩阵特征值	expm1(A)	用 Pade 近似求 e^A
norm(A, 1)	A 阵的 1-范数	expm2(A)	用 Taylor 级数近似求 e^A，精度稍差，但对任何方阵适用
norm(A)	A 阵的 2-范数(A 的最大奇异值)	expm3(A)	用特征值分解求 e^A，仅当独立特征向量数等于秩时适用
norm(A, inf)	A 阵的无穷范	logm(A)	矩阵对数 ln(A)
norm(A, 'fro')	A 阵的 F-范数(A 阵全部奇异值平方和的正开方根)	sqrtm(A)	平方根矩阵 $A^{\frac{1}{2}}$
rank(A)	A 阵的秩(A 阵的非零奇异值数目)		
rcond(A)	矩阵的倒条件数	funm(A, 'FN')	A 阵的一般矩阵函数

【例 1】求矩阵的行列式值、迹、秩、条件数和范数。

```
A=magic(5);
s=svd(A)'
d=det(A),t=trace(A),rk=rank(A),c=cond(A)
n1=norm(A,1),n2=norm(A),ninf=norm(A,inf),nf=norm(A,'fro')
s =
    65.0000     22.5471     21.6874     13.4036     11.9008
d =
    5070000
t =
    65
rk =
     5
c =
    5.4618
n1 =
     65
n2 =
```

```
    65.0000
ninf =
     65
nf =
    74.3303
```

说明:从该例可清楚看到奇异值与矩阵各种性质的关系。

3.5.3 需特别注意区分的两种函数运算

在前面已经用很多篇幅介绍了矩阵运算和数组运算之间的区别。本节要特别强调几种易引起混淆的矩阵函数和数组函数。

数组函数指令	矩阵函数指令
exp(A)	expm(A)
log(A)	logm(A)
sqrt(A)	sqrtm(A)
	funm(A, 'FN')

【例 1】显示 sqrt 与 sqrtm 的区别。

```
B=[7 10;15 22];
Bs=sqrt(B)
BS=sqrtm(B)
BsBs=Bs * Bs
BBs=Bs  * Bs
BSBS=BS * BS
Bs =
    2.6458    3.1623
    3.8730    4.6904
BS =
    1.5667    1.7408
    2.6112    4.1779
BsBs =
   19.2474   23.1990
   28.4129   34.2474
BBs =
    7.0000   10.0000
   15.0000   22.0000
BSBS =
    7.0000   10.0000
   15.0000   22.0000
```

【例 2】求 sin(B) 和 funm(B,'sin')。

```
sinb=sin(B)
fsinmb=funm(B,'sin')
sinb =
    0.6570   -0.5440
    0.6503   -0.0089
fsinmb =
   -0.0272   -0.2410
   -0.3614   -0.3886
```

3.6 关系运算、逻辑运算及其函数

所有的关系运算和逻辑运算都是按数组运算规则定义的。

3.6.1 数组关系运算

实现两个量之间比较的关系运算符如下·

< 小于	<= 小于等于	> 大于
>= 大于等于	== 等于	~= 不等于

运算法则：

(1) 当两个比较量是标量 a 、b 时，则

若 a 、b 间关系成立，那么关系运算结果为“1”；

若 a 、b 间关系不成立，那么关系运算结果为“0”。

(2) 当参与比较的是两个维数相同的数组 A 和 B 时，比较是对 A、B 数组相同位置的元素按标量关系运算规则逐个进行，并给出元素比较结果。最终的关系运算的结果是一个维数与 A(或 B)相同的数组，它的元素由“0”或“1”组成。

(3) 当参与比较的一个是标量 a，而另一个是数组 B 时，则把标量 a 与 B 数组的每一个元素按标量关系运算规则逐个比较，并给出元素比较结果。最终的关系运算的结果是一个维数与 B 相同的数组，它的元素由“0”或“1”组成。

(4) 在算术、关系、逻辑运算中，关系运算的运算“优先权”居中。

【例 1】两种关系运算。

```
A=[1 2 3;4 5 6;7 8 9];
x=5;
X=5*ones(3);
Ax=x>=A
AX=X>=A
```

```
Ax =
     1     1     1
     1     1     0
     0     0     0
AX =
     1     1     1
     1     1     0
     0     0     0
```

结果表明:例中的两种比较结果相同。

3.6.2 数组逻辑运算

逻辑运算符有以下三个:

& 与　　| 或　　~ 非

运算法则:

(1) 在逻辑运算中,确认:

非零元素的逻辑量为"真",用代码"1"表示;

零元素的逻辑量为"假",用代码"0"表示。

(2) 若参与逻辑运算的是两个标量 a 和 b,那么

a&b　　a, b 全为非零时,运算结果为"1";否则为"0"。

a|b　　a, b 中只要有一个非零,运算结果为"1"。

~a　　当 a 是零时,运算结果为"1";当 a 非零时,运算结果为"0"。

(3) 若参与逻辑运算的是两个同维数组 A、B,那么运算将对 A、B 相同位置上的元素按标量规则逐个进行。最终运算结果是一个与 A(或 B)同维的数组,其元素由"1"或"0"组成。

(4) 若参与逻辑运算的一个是标量 a,一个是数组 B,那么运算将在 a 与 B 中的每个元素之间按标量规则逐个进行。最终运算结果是一个与 B 同维的数组,其元素由"1"或"0"组成。

(5) 逻辑"非"是一元运算符,也服从数组运算规则。

(6) 在算术、关系、逻辑运算中,逻辑运算的运算"优先权"最低。

【例 1】数组的三种逻辑运算。

```
A=[1 2 3;4 5 6];
B=[-1 0 0;0 0.5 0];
AandB=A&B
AorB=A|B
notB=~B
AandB =
     1     0     0
     0     1     0
AorB =
```

```
        1       1       1
        1       1       1
notB =
        0       1       1
        1       0       1
```

【例 2】不同运算的优先权。

```
a=1&0+3
b=3>4&1
a1=(1&0)+3
b1=3>(4&1)
a =
        1
b =
        0
a1 =
        3
b1 =
        1
```

说明:用圆括号可改变运算的“优先权”。

3.6.3 关系函数和逻辑函数

从 3 0 版起,MATLAB 就提供了一些关系逻辑函数,而且这种函数的数量随着版本的更新而不断增加。下面是 4 2 版中可调用的的关系逻辑函数。

all 若向量的所有元素非零,则结果为“1”。
any 向量中任何一个元素非零,都给出结果“1”。
exist 检查变量、函数、文件的存在性及类别。
find 找出向量或矩阵中非零元素的位置标识。
finite 若元素值大小有限,则在结果数组的相应位置上标“1”;否则标“0”。
isempty 若被查变量是空阵,则结果为“1”。
isglobal 若被查变量是全局变量,则结果为“1”。
isieee 若计算机采用 IEEE 算法规则,则结果为“1”。
isinf 若元素是“±inf”,则在结果数组的相应位置上标“1”;否则标“0”。
isnan 若元素是“nan”,则在结果数组的相应位置上标“1”;否则标“0”。
issparse 若变量是稀疏矩阵,则在结果数组的相应位置上标“1”;否则标“0”。
isstr 若变量是字符串,则在结果数组的相应位置上标“1”;否则标“0”。
xor 比较两数组元素的“零”、“非零”性质。若元素性质不同,则在结果数组的相应元素位置标“1”,否则标“0”。

这些指令在 MATLAB 程序或直接交互运算中非常有用。考虑到用户可以从用户指南或

在线帮助中很容易地查到具体调用格式，因此本书不介绍具体指令，而只给一些算例。

【例 1】指令 finite, isinf, isnan, isstr 用法示例。

```
X=[-2,-1,0,1,2];
X1=1./X
X2=0./X
X1f=finite(X1)              %X1 中哪些元素是有限的
X1inf=isinf(X1)             %X1 中哪些元素是无限的
X2nan=isnan(X2)             %X2 中哪些元素是“非数 NaN”
X1s=isstr(X1)               %X1 是字符串吗
X2s=isstr(X2)               %X2 是字符串吗
Warning: Divide by zero
X1 =
   -0.5000   -1.0000      Inf    1.0000    0.5000
Warning: Divide by zero
X2 =
     0     0   NaN     0     0
X1f =
     1     1     0     1     1
X1inf =
     0     0     1     0     0
X2nan =
     0     0     1     0     0
X1s =
     0
X2s =
     0
```

说明：

(1) MATLAB 采用 IEEE 浮点算法规则。因此，在 MATLAB 中，除以零的运算也是合法的。本例运算所得 X1 中的“Inf”是“1/0”的结果；而 X2 中的“NaN”是“0/0”的结果。

(2) X1s、X2s 分别说明 X1、X2 都不是字符串变量。

【例 2】指令 find 的用法示例。

```
fX1=find(X1)
fX1g=find(abs(X1)>0 1&abs(X1)<0 6)
fX2=find(X2)
fX20=find(X2==0)
fX2g=find(X2>0)
fX1 =
```

```
     1     2     3     4     5
fX1g =
     1     5
fX2 =
     []
fX20 =
     1     2     4     5
fX2g =
     []
```

【例 3】指令 all 及 any 使用示例。

```
A=magic(6);                %利用 magic 指令产生一个 6 阶魔方阵
A=A(1:3,:);                %仅取 6 阶魔方阵的上三行
A(:,3)=zeros(3,1)          %使 A 阵第三列为全零列
A1all=all(A(:,1)>5)        %A 阵的第一列元素全大于 5 吗
Aall=all(A>5)              %A 阵中哪些列的元素全大于 5
A1any=any(A(:,1)>5)        %A 阵第一列中有大于 5 的元素吗
Aany=any(A>5)              %A 阵哪些列有大于 5 的元素
A =
    35     1     0    26    19    24
     3    32     0    21    23    25
    31     9     0    22    27    20
A1all =
     0
Aall =
     0     0     0     1     1     1
A1any =
     1
Aany =
     1     1     0     1     1     1
```

3.7 矩阵分解函数

MATLAB 的矩阵分解函数有些是内部函数(Built－in Functions),有些则是外部的函数文件(External Files),还有些存在于各种各样的工具包里。表 3.7-1 是基本的矩阵分解函数。

表 3.7-1 矩阵分解函数表

分解指令	含 义	分解指令	含 义
cdf2rdf	把复数对角形转换成实数块对角形	qr	QR 正交三角分解
chol	Cholesky 分解	qz	QZ 分解(用于广义特征值)
eig	矩阵的特征值分解	rref	转换为行阶梯形式
hess	转换为 Hessenberg 形式	rsf2csf	把实数块对角形转换成复数对角形
lu	LU 三角分解	schur	Schur 分解
null	零空间	subspace	子空间夹角
orth	值空间	svd	奇异值分解
pinv	伪逆		

下面将比较详细地介绍特征值分解指令和奇异值分解指令,其他指令请查阅在线帮助。

3.7.1 特征值分解

在这一小节中着重介绍有满秩独立特征向量系的方阵的特征值的求解问题。至于独立特征向量系不满秩的特征值分解问题将稍后介绍。这是因为在数值计算中特征根为重根的情况极少发生。关于这种重根情况的详细讨论将放在第四章。

本节主要介绍指令 eig 和 cdf2rdf 。

普通特征值问题

对于特征值问题:Ax=λx,有以下几种调用格式可用来求 A 阵的特征值和特征向量:

d=eig(A)

[V,D]=eig(A)

[V,D]=eig(A,'nobalance')

其中,A 是方阵,且它的独立特征向量系满秩;d 是 A 阵特征值排成的列向量;D 是 A 阵的特征值对角阵,V 由 A 阵的全部右特征向量构成,且有 A * V=V * D ;“'nobalance'”用在 A 阵中有的元素小到与截断误差相当时。

【例 1】简单实阵的特征值问题。

```
A=[1,-3;2,2/3];
[V,D]=eig(A)
V =
  -0.7728 + 0.0527i    -0.7728 - 0.0527i
        0 + 0.6325i          0 - 0.6325i
D =
   0.8333 + 2.4438i          0
        0               0.8333 - 2.4438i
```

【例2】当矩阵中有的元素与截断误差相当时，特征值问题的讨论。

```
A=[3        -2       -0.9     2*eps
   -2       4        -1       -eps
   -eps/4   eps/2    -1       0
   -0 5     -0 5     0.1      1 ];
[V1,D1]=eig(A);
ER1=A*V1-V1*D1
[V2,D2]=eig(A,'nobalance');
ER2=A*V2-V2*D2
ER1 =
    0.0000         0    0.0000    0.0000
         0    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    1.2896    0.0000
ER2 =
  1.0e-014 *
   -0.1332    0.0444   -0.0255   -0.0555
    0.2665    0.1110    0.0219    0.0611
   -0.0053   -0.0111    0.0106         0
    0.0472   -0.0222    0.0222    0.0056
```

说明：

(1) 在本例情况下，若求特征值指令中不采用“'nobalance'”，那么所得结果是有严重计算错误的。这是由于在执行eig指令过程中首先要调用使原矩阵各元素大致相当的“平衡”程序。该“平衡”程序使得原方阵中本可忽略的小元素的作用被放大了。

(2) 当A阵来自程序的中间计算结果，且在A的结果显示中含有形式“零”元素时，就有可能发生本例的情况。

(3) 但从一般情况说，“平衡”程序的作用是减小计算误差。

复数特征值对角阵与实数块特征值对角阵的转化

即使是实阵，其特征值中也可能出现复数。而在实际使用中，又常需要把这一对对共轭复数特征值转化为一个个(2×2)的实数块。为此，MATLAB提供了下面两个指令。

[VR,DR]=cdf2rdf(VC,DC)　　　%把复数对角形转换成实数块对角形

[VC,DC]=rsf2csf(VR,DR)　　　%把实数块对角形转换成复数对角形

其中，DC是含复数的特征值对角阵，VC是与之相应的特征向量阵；DR是含实数块的“特征值”对角阵，VR是与之相应的实“特征向量”阵。

【例3】把例1中的复数特征值对角阵D转换成实数块对角阵。

```
[VR,DR]=cdf2rdf(V,D)
```

```
VR =
   -0.7728        0.0527
         0        0.6325
DR =
    0.8333        2.4438
   -2.4438        0.8333
```

说明：VR * DR/VR＝A，有兴趣的读者可以检验。

广义特征值问题

这里所说的广义特征值问题是指：$Ax=\lambda Bx$ 的非平凡解问题。式中 A，B 都是同维的方阵。解这类问题的指令调用格式是

d＝eig(A，B)

[V，D]＝eig(A，B)

这里：D 是广义特征值对角阵；V 是相应的广义特征向量阵，且 A * V＝B * V * D 。

3.7.2　奇异值分解和伪逆

奇异值分解

奇异值分解在矩阵分析中占有极为重要的位置。奇异值分解的定义是：

对矩阵 $A\in C^{m\times n}$(不失一般，设 $m>n$)，若存在酉阵 $U\in C^{m\times m}$，$V\in C^{n\times n}$使

$$A = U\Sigma V^T \tag{1}$$

其中，$\Sigma=\begin{bmatrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ 0 & \cdots & \\ \cdot & & \vdots \\ 0 & \cdots & 0\end{bmatrix}_{m\times n}$，$\sigma_1\geqslant\sigma_2\geqslant\cdots\geqslant\sigma_n\geqslant0$，那么，U，$\Sigma$，V 称为矩阵 A 的奇异值分解三对组。这里的上标“T”表示共轭转置。

上述分解表达式可更简洁地写成：

$$A = U_0\Sigma_0V^T \tag{2}$$

其中，$\Sigma_0=\mathrm{diag}(\sigma_1,\cdots,\sigma_n)$。

MATLAB 提供的奇异值分解函数 svd 是利用 LINPACK 程序库中的 ZSVDC 编制而成的，假如经 75 步 QR 迭代仍找不到一个奇异值，那么将给出“不收敛”的提示。奇异值分解指令的调用格式如下：

s＝svd(A)

[U，S，V]＝svd(A)

[U0，S0，V]＝svd(A，0)

假设 A 阵的行数大于列数，即 m>n ，那么 s 是 A 阵 n 个奇异值所构成的列向量；S 是式

(1)中的Σ，S0 是式(2)中的Σ_0；其余各量都与上面两式中的符号对应；第三个指令调用格式中的第二个输入宗量“0”(零)，使 svd 给出简洁分解。

【例 1】奇异值分解。

```
A=[0.96,1.72;2.28,0.96];
[U,S,V]=svd(A)
U =
   -0.6000    0.8000
    0.8000   -0.6000
S =
    3.0000         0
         0    1.0000
V =
    0.8000   -0.6000
    0.6000    0.8000
```

伪逆

在 MATLAB 中，求 Moore－Penrose 伪逆(Pseudoinverse)的指令调用格式如下：

X=pinv(A)

X=pinv(A, tol)

这两个指令的差别仅在于是否由用户自己指定为零奇异值的阈值 tol 。假如不指定，那么，缺省地认为 tol=max(size(A))＊norm(A)＊eps 。

当用伪逆去求解列不满秩超定最小二乘问题 Ay=b 时，它所给出的解 $\hat{y}$ 满足

$\|A\hat{y}-b\| = \min\|Ay-b\|$

即最小范数最小二乘解。

【例 2】两种最小二乘解的对比。

```
A=magic(8);A=A(:,1:6);            %构造(8×6)矩阵
b=260*ones(8,1);                  %构造(8×1)向量
yp=(pinv(A)*b)'
yd=(A\b)'
nyp=norm(yp)
nyd=norm(yd)
yp =
    1.1538    1.4615    1.3846    1.3846    1.4615    1.1538
Warning: Rank deficient, rank = 3 tol = 1.8829e-013
yd =
    3.0000    4.0000         0         0    1.0000         0
nyp =
```

```
        3.2817
nyd =
        5.0990
```

说明：

（1）由伪逆求得的最小二乘解的范数最小。

（2）由除法求得的最小二乘解的非零元素数等于 A 阵的秩数。

3.8 向量和矩阵处理

在前面几节里，实际上已经用到了许多向量和矩阵的操作处理（Manipulation）指令。本节将系统介绍向量的生成、标识（Subscripting）的使用、空阵（Empty Matrix）的作用、常用矩阵（Elementary Matrices）和特殊矩阵（Specialized Matrices）的生成、矩阵结构的变化（Matrix Manipulation）。

3.8.1 向量的生成

利用冒号生成行向量

冒号“：”的使用格式如下：

（1）x = N1:N2

必须有 N2≥N1；生成行向量 x = [N1, N1 + 1, N1 + 2, …, N1 + k]，在此 k 满足 N2 − 1 < N1 + k ≤ N2 。

（2）x = N1:dn:N2

当 dn>0 时，必须有 N2≥N1。生成行向量 x = [N1, N1 + dn, N1 + 2 × dn, …, N1 + k × dn]，在此 k 满足 N2 − dn < N1 + k × dn ≤ N2 。

当 dn<0 时，必须有 N2≤N1。生成行向量 x = [N1, N1 + dn, N1 + 2 × dn, …, N1 + k × dn]，在此 k 满足 N2 ≤ N1 + k × dn < N2 − dn。

在以上使用格式中 k 是自然数。

【例 1】冒号生成行向量。

```
x1=0:10         %可形成由0到10，间隔为1的行向量
x2=0:3:11       %形成由0到不大于11，间隔为3的行向量
x3=11.5:-3:0
x1 =
     0     1     2     3     4     5     6     7     8     9    10
x2 =
     0     3     6     9
x3 =
```

```
11.5000 8.5000 5.5000 2.5000
```

利用线性等分指令生成向量

线性等分指令的调用格式是：

y = linspace(x1, x2)　　生成(1×100)维行向量，且 y(1) = x1，y(100) = x2。

y = linspace(x1, x2, n)　　生成(1×n)维行向量，且 y(1) = x1，y(n) = x2。

冒号指令和线性等分指令都能产生线性等分向量。但前者是通过设定间隔去生成维数随之确定的等间隔向量，而后者是通过设定向量维数去生成等间隔向量。

MATLAB 3 0 版没有该指令。

利用对数等分指令生成向量

在自动控制、数字信号处理中常需要对数刻度，MATLAB 提供了对数等分指令。它的调用格式为：

y = logspace(a, b)　　生成(1×50)维对数等分行向量，且 y(1) = 10^a，y(50) = 10^b。

y = logspace(a, b, n)　　生成(1×n)维对数等分行向量，且 y(1) = 10^a，y(n) = 10^b。

y = logspace(a, pi)　　生成(1×50)维对数等分行向量，且 y(1) = 10^a，y(50) = π。

【例 2】生成向量的三种指令对比。

```
y1 = 1:200:1000
y2 = linspace(1,1000,5)
y3 = logspace(0,3,5)
y1 =
     1      201      401      601      801
y2 =
   1.0e+003 *
     0.0010    0.2507    0.5005    0.7502    1.0000
y3 =
   1.0e+003 *
     0.0010    0.0056    0.0316    0.1778    1.0000
```

3.8.2 标　识

矩阵的元素、子阵可以通过标量、向量、冒号的标识来援引和赋值。

(1) 矩阵元素的标识方式 A(ni, nj)

ni，nj 都是标量。若它们不是整数，则在调用格式中会自动圆整到最邻近整数。ni 指定元素的行位置；nj 指定元素的列位置。

(2) 子阵的序号向量标识方式 A(v, w)

v，w 可以是任何排序的向量。推荐使用序号单调向量，否则可能产生混乱。v，w 中任一个可以是冒号“:”，它表示取全部行(在 v 位置)或列(在 w 位置)。

v , w 中所用序号必须大于等于 1 且小于等于矩阵维数。否则将失败。

(3) “0—1”向量标识方式 A(L1, :)、A(:, L2)、A(L1, L2)

向量 L1、L2 的长度分别等于矩阵 A 的行维、列维；向量 L1、L2 中的元素或取 1（表示提取相应的行或列）或取 0（不提取）；作这种标识方法主要与关系运算配合使用。

【例 1】元素和子阵的标识。

```
B=[1 2 3 4 5;6 7 8 9 10;11 12 13 14 15]
B23=B(2,3)
B1=B(1:2,[1 3 5])
B2=B([3 1],:)
B([1 3],[2 4])=zeros(2)
B =
     1     2     3     4     5
     6     7     8     9    10
    11    12    13    14    15
B23 =
     8
B1 =
     1     3     5
     6     8    10
B2 =
    11    12    13    14    15
     1     2     3     4     5
B =
     1     0     3     0     5
     6     7     8     9    10
    11     0    13     0    15
```

【例 2】“0 - 1”向量标识。

```
rand('seed',0)          %设定随机种子是为了仿真计算有重复性
X=rand(1,5)             %产生(1×5)均布随机向量
L=X<=0.5                %标出小于等于 0 5 的元素位置
X=X(L)                  %获得元素都不超过 0 5 的子向量
X =
    0.2190    0.0470    0.6789    0.6793    0.9347
L =
   1    1    0    0    0
X =
    0.2190    0.0470
```

3.8.3 空 阵

在 MATLAB 中定义“[ ]”为空阵。向一个变量赋空阵的语句为：

X=[ ]

一个被赋予空阵的变量(如 X)具有以下性质：

(1) 在 MATLAB 工作内存中确实存在被赋空阵的变量 X。这可以用下列方式来验证：

(A)在 MATLAB 中，“敲入”X，运行后，屏上将给出，“X=[]”的显示。

(B)“敲入” exist('X')，运行后将给出“1”，表明 X 确实存在。

(C)“敲入” isempty(X)，运行后将给出“1”，表明 X 确实是空阵。

(D)“敲入” size(X)，运行结果是“0 0”。

(E)“敲入” whos，从内存变量表中可看到，X 确实存在，且 X 是(0×0)矩阵。

(2) 空阵中不包含任何元素，它的维数是(0×0)。

(3) 空阵可以在 MATLAB 的运算中传递。

(4) 可以用 clear 从内存中清除空阵变量。

要特别注意：空阵不是“零”；空阵也不是“不存在”；空阵也不具备“clear”清除变量的作用。空阵主要用来使矩阵按要求缩维。

【例 1】空阵的缩维作用。

```
A=[1 2 3 4 5 6;7 8 9 10 11 12;13 14 15 16 17 18]
AI=A(:,[1 3 4 6])
A(:,[2 5])=[]
A =
     1     2     3     4     5     6
     7     8     9    10    11    12
    13    14    15    16    17    18
AI =
     1     3     4     6
     7     9    10    12
    13    15    16    18
A =
     1     3     4     6
     7     9    10    12
    13    15    16    18
```

结果表明：用空阵缩维后的 A 阵与借助向量标识而得的 AI 阵相同。

3.8.4 常用矩阵的生成

MATLAB 为方便编程和运算，提供的常用矩阵生成指令有：

eye	单位阵	**ones**	全 1 阵	**zeros**	全零阵
rand	均布随机阵	**randn**	正态随机阵		
linspace	线性等分向量	**logspace**	对数等分向量	**meshgrid**	

(lınspace 和 logspace 见第 3 8 1 小节。meshgrıd 用于构作网面图,将在第五章介绍。)

单位阵、全 1 阵、全零阵的生成

产生单位阵、全 1 阵、全零阵的指令调用格式相同,有以下几种:

eye(n)　　产生(n×n)维单位阵。
ones(n)　　产生(n×n)维全 1 阵。
zeros(n)　　产生(n×n)维全 0 阵。
eye(m, n)　　产生(m×n)维单位阵。
ones(m, n)　　产生(m×n)维全 1 阵。
zeros(m, n)　　产生(m×n)维全 0 阵。
eye(size(A))　　产生与 A 同维的单位阵。
ones(size(A))　　产生与 A 同维的全 1 阵。
zeros(size(A))　　产生与 A 同维的全 0 阵。

注意:MATLAB 老版本中,产生与 A 同维矩阵的指令 eye(A) , ones(A) , zeros(A) , 在新版中已被更为严谨的 eye(sıze(A)), ones(sıze(A)), zeros(size(A)) 所替代。如用户仍用老指令,那么新版 MATLAB 将给出警告和建议,但老指令仍被执行。

随机阵的生成

在新老版本中产生随机阵的指令有较大不同。在此以介绍新版指令为主,对老指令也予以适当的说明。

(1) 在[0, 1]区间的均匀分布随机数

rand(n)　　产生(n×n)维均布随机阵。
rand(m, n)　　产生(m×n)维均布随机阵。
rand(sıze(A))　　产生与 A 同维的均布随机阵。
rand('seed', a)　　令 a 为随机发生器的种子。
rand('seed')　　获取随机发生器的当前种子值。

(2) 服从 N(0, 1)分布的正态随机数(有关算例见 3.10 节的例)

randn(n)　　产生(n×n)维正态随机阵。
randn(m, n)　　产生(m×n)维正态随机阵。
randn(sıze(A))　　产生与 A 同维的正态随机阵。
randn('seed', a)　　令 a 为随机发生器的种子。
randn('seed')　　获取随机发生器的当前种子值。

(3) 在 MATLAB 的早期版本中,随机指令只有一个“rand”。均布随机数和正态随机数的获取靠字符串作如下切换:

rand('normal')　　把当前随机发生器切换到“正态”发生工作状态。
rand('unıform')　　把当前随机发生器切换到“均布”发生工作状态。

MATLAB 默认“均布”为随机发生器的缺省工作状态。在 4.0 版以上的 MATLAB 中，这种切换指令已被废除。

【例 1】在十万个[0,1]间均布随机数中，小于等于 0.5 的随机数有多少个？

```
rand('seed',0)
X=rand(1,100000);
N05=sum(X<=0.5)
N05 =
        49816
```

说明：

(1) 第一句把均布随机发生器的“种子”置零，即 MATLAB 随机发生器的初始状态。

(2) 采用种子设定随机发生器的工作状态的目的是，此后所产生的随机数具有可重复性。在许多仿真实验中，需要采取这一措施。

(3) 指令“X<=0.5”形成一根长为十万的“0-1”行向量。原 X 向量中大于 0.5 的元素都变成了 0，其余的都变成 1。

(4) sum 指令求出“0-1”向量所有元素的和，也即 X 向量中小于等于 0.5 的元素数目。

3.8.5 特殊矩阵的生成

MATLAB 给出一组在线性代数、信号处理及控制中经常遇到的特殊矩阵的生成指令：

compan	伴随阵	**diag**	对角阵
gallery	试验矩阵	**hadamard**	Hadamard 阵
hankel	Hankel 阵	**hilb**	Hilbert 阵
invhilb	逆 Hilbert 阵	**kron**	Kronecker 张量积
magic	魔方阵	**pascal**	Pascal 三角阵
toeplitz	Toeplitz 阵	**vander**	Vandermonde 阵
wilkinson	Wilkinson 特性值试验阵		

本节只介绍求 Kronecker 张量积的指令；其余指令请看用户指南，或用 help 获得在线帮助。

Kronecker 张量积

Kronecker 积的数学定义是

$$A_{mn} \oplus B_{pq} = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \vdots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}_{(m\times p)\times(n\times q)}$$

MATLAB 提供的求 Kronecker 积的指令调用格式如下：

K=kron(A,B)

【例 1】求 Kronecker 积。

```
A=[1 2 3;4 5 6]
B=[2 4;6 8]
C=kron(A,B)
A =
        1       2       3
        4       5       6
B =
        2       4
        6       8
C =
        2       4       4       8       6      12
        6       8      12      16      18      24
        8      16      10      20      12      24
       24      32      30      40      36      48
```

3.8.6 矩阵结构变换

矩阵的旋转、翻转

B=rot90(A)　　　B 由 A 逆时针方向旋转 90°而得。

B=rot90(A,k)　　B 由 A 逆时针方向旋转(k×90°)而得，k 可取正负整数。

B=fliplr(A)　　　B 由 A 阵左右翻转而得。

B=flipud(A)　　　B 由 A 阵上下翻转而得。

上述指令都是函数文件，文件的核心是向量标识技术。MATLAB 3 0 版无翻转指令。

【例 1】矩阵的旋转和转置的区别。

```
A=[1 2 3 4;5 6 7 8;9 10 11 12]
B1=rot90(A)
BT=A'
B2=rot90(A,2)
A =
        1       2       3       4
        5       6       7       8
        9      10      11      12
B1 =
        4       8      12
        3       7      11
        2       6      10
        1       5       9
```

```
BT =
     1     5     9
     2     6    10
     3     7    11
     4     8    12
B2 =
    12    11    10     9
     8     7     6     5
     4     3     2     1
```

矩阵的变维

实现变维主要有两种方式:“:”和“reshape”。前者比较简单、隐晦;后者比较明确完善。在此主要介绍后者。

B=reshape(A,m,n)

说明:

(1) 该指令所得 B 阵的维数将是(m×n)。注意:(m×n)必须等于 A 阵的列维和行维之积。

(2) 在变维中,元素按列重新排序。

(3) MATLAB 3 0 版无此指令。

【例 2】变维示例。

```
A=[1 2 3 4;5 6 7 8;9 10 11 12];
B=reshape(A,2,6)
C=zeros(2,6);
C(:)=A(:)
B =
     1     9     6     3    11     8
     5     2    10     7     4    12
C =
     1     9     6     3    11     8
     5     2    10     7     4    12
```

说明:

(1) 第三条指令先给 C 定维。

(2) 等号右边的“A(:)”表示 A 阵将以列的排序送出它的元素。

(3) 等号左边的“C(:)”表示 C 阵将以列的排序接受送来的元素。

矩阵部分元素的抽取

v=diag(A,k)　　v 由 A 阵第 k 条对角线的元素组成。0 为主对角线;k 取正整数为上第 k 对角线;k 取负整数为下第 k 对角线。

v=diag(A)	v 由 A 阵主对角线元素组成。
A=diag(v,k)	A 阵的第 k 条对角线元素取自向量 v 。k 的意义同上。
A=diag(v)	A 阵的主对角线元素取自向量 v 。
L=tril(A,k)	L 第 k 条对角线及以下元素取 A 阵相应元素,L 阵其余元素都为零。
L=tril(A)	L 主对角线及以下元素取 A 阵相应元素,L 阵其余元素都为零。
U=triu(A,k)	U 第 k 条对角线及以上元素取 A 阵相应元素,U 阵其余元素都为零。
U=triu(A)	U 主对角线及以上元素取 A 阵相应元素,U 阵其余元素都为零。

【例 3】部分元素抽取。

```
A=[1 2 3 4;5 6 7 8;9 10 11 12]
v=diag(A)
LA=tril(A,-1)
UA=triu(A,1)
V=diag(v)
A =
     1     2     3     4
     5     6     7     8
     9    10    11    12
v =
     1
     6
    11
LA =
     0     0     0     0
     5     0     0     0
     9    10     0     0
UA =
     0     2     3     4
     0     0     7     8
     0     0     0    12
V =
     1     0     0
     0     6     0
     0     0    11
```

3.8.7 矩阵的扩展

第 3 8.3 小节介绍了如何利用空阵使矩阵的维数收缩。本小节介绍两种矩阵扩展技巧。
(1) 利用对矩阵标识块赋值指令:X(m1:m2,n1:n2)=A ,生成大矩阵。(m2-m1+1)必

须等于 A 的行维。(n2－n1＋1)必须等于 A 的列维。生成的(m2×n2)维 X 矩阵除赋值子阵和原已存在的元素外，其余元素都取零。

(2) 利用方括号和小矩阵生成大矩阵。第 3.1.3 节中的例 3 就利用了这种技巧。下面再举一例。

【例 1】用三种方法生成多项式 S^3-7S+6 的伴随阵。

```
CP=[1 0 -7 6];
A1=compan(CP)
A2=[-CP(2:4);eye(2) zeros(2,1)]
A3=-CP(2:4);A3(2:3,1:2)=eye(2)
A1 =
      0      7     -6
      1      0      0
      0      1      0
A2 =
      0      7     -6
      1      0      0
      0      1      0
A3 =
      0      7     -6
      1      0      0
      0      1      0
```

3.9 多项式

3.9.1 多项式的表达和创建

多项式表达方式的约定

多项式 $P(x)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n$ 用以下系数行向量表示：

$$P=[a_0 \quad a_1 \cdots a_{n-1} \quad a_n]$$

多项式行向量的创建方法

(1) 多项式系数向量的直接输入法。

(2) 利用指令：P＝poly(AR)，产生多项式系数向量。若 AR 是方阵，则多项式为特征多项式；若 AR 是向量，即 $AR=[ar_1 \quad ar_2 \quad \cdots \quad ar_n]$，则所得的多项式满足关系式：

$$(x-ar_1)(x-ar_2)\cdots(x-ar_n)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}+a_n$$

【例 1】求 3 阶方阵 A 的特征多项式。

```
A=[11 12 13;14 15 16;17 18 19];
PA=poly(A)
PPA=poly2str(PA,'s')
PA =
     1.0000     -45.0000     -18.0000      0.0000
PPA =
    s^3 - 45 s^2 - 18 s + 4.264e-014
```

注意：

(1) n 阶方阵的特征多项式系数向量一定是(n+1)维的。

(2) 特征多项式系数向量的第一个元素必是 1。

【例 2】由给定根向量求多项式系数向量。

```
R=[-0.5,-0.3+0.4*i,-0.3-0.4*i];
P=poly(R)、
PR=real(P)
PPR=poly2str(PR,'x')
P =
    1.0000    1.1000    0.5500 + 0.0000i    0.1250
PR =
    1.0000    1.1000    0.5500    0.1250
PPR =
    x^3 + 1.1 x^2 + 0.55 x + 0.125
```

说明：

(1) 要形成实系数多项式，根向量中的复数根必须共轭成对。

(2) 含复数根的根向量所生成的多项式系数向量(如 P)的系数有可能带在截断误差数量级的虚部。此时可采用取实部的指令“real”把虚部滤掉。

(3) poly2str 是一个函数文件，它存在于 MATLAB 的控制工具包(Control Toolbox)中。

3.9.2 多项式乘除运算

卷积和多项式乘运算

长度为 m 的向量序列 a 和长度为 n 的向量序列 b 的卷积(Convolution)c 定义为：

$$c(k) = \sum_{j=1}^{k} a(j)b(k+1-j)$$

式中，c 向量序列的长度为(m+n-1)。而求两个多项式的积(Polynomial mutiplication product)的机理与卷积完全相同。

在 MATLAB 中实现以上两种运算的指令是：c=conv(a, b)。

【例 1】展开$(s^2+2s+2)(s+4)(s+1)$。

```
c=conv([1,2,2],conv([1,4],[1,1]))
C=poly2str(c,'s')
c =
      1      7      16      18      8
C =
s^4 + 7 s^3 + 16 s^2 + 18 s + 8
```

解卷和多项式除运算

解卷(Deconvolution)是卷积的逆运算。更详细地说，用向量 a 对向量 c 进行解卷将得到“商(Quotient)”向量 q 和“余(Remainder)”向量 r，并且满足

$$c(k)-r(k)=\sum_{j=1}^{k}a(j)q(k+1-j)$$

从数学上讲，以上解卷运算也可被理解为：多项式 c 除以多项式 a 将得到商多项式 q 和余多项式 r。该运算的指令调用格式是：

[q, r]=deconv(c, a)

【例 2】求例 1 所得向量 c 分别被(s+4)、(s+3)除后的结果。

```
[q1,r1]=deconv(c,[1,4])
[q2,r2]=deconv(c,[1,3])
cc=conv(q2,[1,3])
test=((c-r2)==cc)
q1 =
      1      3      4      2
r1 =
      0      0      0      0      0
q2 =
      1      4      4      6
r2 =
      0      0      0      0     -10
cc =
      1      7      16     18     18
test =
      1      1      1      1      1
```

说明：

(1) $C=s^4+7s^3+16s^2+18s+8$ 不能被(s+3)整除，余项是常数(-10)。

(2) 检验结果表明运算正确。

3.9.3 常用多项式运算指令

r = roots(p)	求多项式向量 p 的根。
PA = polyval(p, S)	按数组运算规则计算多项式值。p 为多项式，S 为矩阵。
PM = polyvalm(p, S)	按矩阵运算规则计算多项式值。p 为多项式，S 为矩阵。
[r, p, k] = residue(b, a)	部分分式展开。 b 、a 分别是分子、分母多项式系数向量； r 、p 、k 分别是留数、极点、直项向量。
P = polyfit(x, y, n)	用 n 阶多项式拟合 x ，y 向量给定的数据。

说明：对于多项式 b(s)与不含重根的 na 阶多项式 a(s)之比，如下展开称为部分分式展开：

$$\frac{b(s)}{a(s)} = \frac{r_1}{s-p_1} + \frac{r_2}{s-p_2} + \cdots + \frac{r_n}{s-p_n} + k(s)$$

式中，$p_1, p_2, \cdots, p_n$ 称为极点(Poles)，$r_1, r_2, \cdots, r_n$ 称留数(Residues)，k(s)称为直项(Direct term)。假如 a(s)含有 m 重根 p_j，那么相应部分则写为：

$$\frac{r_j}{s-p_j} + \frac{r_{j+1}}{(s-p_j)^2} + \cdots + \frac{r_{j+m-1}}{(s-p_j)^m}$$

【例 1】求$(x^3-6x^2-72x-27)$的根。

```
r=roots([1, -6, -72, -27])
r =
    12.1229
    -5.7345
    -0.3884
```

说明：

(1) MATLAB 在求多项式根时，并不按经典方法直接对多项式进行运算，而是先把多项式转化为伴随阵，然后再求特征值。理论已经证明，这种求多项式根的现代方法的计算可靠性及精度都高于经典方法。

(2) 请注意，MATLAB 约定：多项式系数用行向量表示，一组根用列向量表示。

【例 2】两种多项式求值指令的差别。

```
S=pascal(4)              %生成一个 4 阶矩阵
P=poly(S);
PP=poly2str(P,'s')
PA=polyval(P,S)          %独立变量取数组 S 元素时的多项式值
PM=polyvalm(P,S)         %独立变量取矩阵 S 时的多项式值
S =
     1     1     1     1
     1     2     3     4
```

```
      1      3      6     10
      1      4     10     20
PP =
   s^4 - 29 s^3 + 72 s^2 - 29 s + 1
PA =
  1.0e+004 *
     0.0016     0.0016     0.0016     0.0016
     0.0016     0.0015    -0.0140    -0.0563
     0.0016    -0.0140    -0.2549    -1.2089
     0.0016    -0.0563    -1.2089    -4.3779
PM =
  1.0e-010 *
    -0.0027    -0.0094    -0.0222    -0.0428
    -0.0094    -0.0326    -0.0749    -0.1423
    -0.0222    -0.0749    -0.1713    -0.3233
    -0.0428    -0.1423    -0.3233    -0.6091
```

说明：PM中的元素都很小，它是运算误差造成的。从理论上讲，它应该为零。这就是著名的"Caylay－Hamilton"定理：任何一个矩阵满足它自己的特征多项式方程。

【例3】部分分式展开。

```
a=[1,3,4,2,7,2];
b=[3,2,5,4,6];
[r,s,k]=residue(b,a)
r =
   1.1274 + 1.1513i
   1.1274 - 1.1513i
  -0.0232 - 0.0722i
  -0.0232 + 0.0722i
   0.7916 + 0.0000i
s =
  -1.7680 + 1.2673i
  -1.7680 - 1.2673i
   0.4176 + 1.1130i
   0.4176 - 1.1130i
  -0.2991
k =
     []
```

说明：

(1) 当分母阶数高于分子时，计算结果中的k将是空阵，表示无此项。

(2) 从计算数学上讲，在一些根很靠近时，极点和留数的计算对截断误差的反映是很灵敏的。此时，这种表达方式不如状态方程或零、极点展开可靠。

【例 4】用 6 阶多项式对区间 [0,2 5]上的误差函数 $y(x)=\frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt$ 进行最小二乘拟合。

```
x=0:0 1:2 5;
y=erf(x);                          %计算“误差函数”在 [0,2 5] 内的数据点
P=polyfit(x,y,6)
Px=poly2str(P,'x')
P =
    0.0084   -0.0983    0.4217    -0.7435    0.1471    1.1064    0.0004
Px =
   0.008419 x^6 - 0.0983 x^5 + 0.4217 x^4 - 0.7435 x^3 + 0.1471 x^2 + 1.106 x
+ 0.0004412
```

【例 5】有效拟合的区间性图示。

```
x=0:0.1:5;
y=erf(x);
f=polyval(P,x);
plot(x,y,'bo',x,f,'r-')
axis([0,5,0,2])
legend('拟合曲线','原数据线')
```

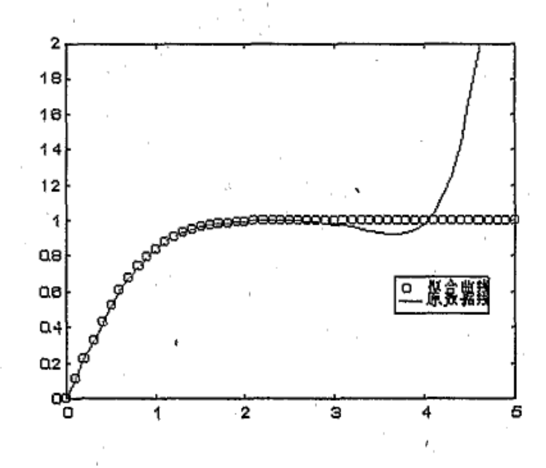


图 3 9-1 有效拟合的区间性

说明：从图 3.9-1 可见，在 [0,2.5] 区间之外，两条曲线的表现完全不同。

3.10 数据分析

MATLAB 对数据分析指令的两条约定是：

(1) 若输入宗量 X 是向量，那么不管是行向量还是列向量，运算是对整个向量进行的。

(2) 若输入宗量 X 是数组(或称矩阵)，那么指令运算是按列进行的。即默认每个列是由一个变量的不同“观察”所得的数据组成。

这两条约定不仅应用于本节所提到的指令，而且反映在 MATLAB 各工具包的软件中。为此向用户建议：在编制自己的软件时，请尽量遵循以上约定，以便更好地利用 MATLAB 现有的函数和指令。

3.10.1 基本统计指令

max(X)	找 X 各列的最大元素。
mean(X)	求 X 各列的平均值。
median(X)	找 X 各列的中位元素。
min(X)	找 X 各列的最小元素。
std(X)	找 X 各列的标准差。
prod(X)	找 X 各列元素之积。
sum(X)	找 X 各列元素之和。
S=cumsum(X)	求 X 各列元素累计和(可用于积分计算)。S 与 X 同维，且 $s_{ij}=\sum_{k=1}^{i}x_{kj}$。
P=cumprod(X)	求 X 各列元素累计积。P 与 X 同维，且 $p_{ij}=\prod_{k=1}^{i}x_{kj}$。
sort(X)	使 X 各列元素按递增排序。

【例 1】做基本统计。

```
randn('seed',0)
A=randn(100,4);
AMAX=max(max(A))
AMED=median(A)
AMEAN=mean(A)
ASTD=std(A)
AMAX =
      3.2069
AMED =
      0.1489      0.0301      0.0443      -0.0189
AMEAN =
      0.0727      0.0264      0.0351       0.0424
```

```
ASTD =
        0.9536      1.0628      1.0860        0.9927
```

3.10.2 协方差阵和相关阵

C=cov(X)　　求协方差阵 $\mathrm{cov}\{X\}=E\{(X-\mu_x)^T(X-\mu_x)\}$。

C=cov(x, y)　　求两随机变量的协方差 $\begin{bmatrix} \sigma_x^2 & \mathrm{cov}(x,y) \\ \mathrm{cov}(y,x) & \sigma_y^2 \end{bmatrix}$。

P=corrcoef(X)　　求相关阵，且 $p(i,j)=\dfrac{C(i,j)}{\sqrt{C(i,i)}\sqrt{C(j,j)}}$。

P=corrcoef(x, y)　　求两随机变量的(2×2)相关系数。

【例 1】两组指令在随机阵 X, Y 上的例子。

```
rand('seed',1)
X=rand(10,3);
Y=rand(10,3);
CX=cov(X)
CY=cov(Y)
Cxy=cov(X,Y)
PX=corrcoef(X)
Pxy=corrcoef(X,Y)
CX =
      0.0508     -0.0159      0.0442
     -0.0159      0.0332      0.0006
      0.0442      0.0006      0.0882
CY =
      0.0870      0.0656     -0.0318
      0.0656      0.0874     -0.0576
     -0.0318     -0.0576      0.1013
Cxy =
      0.0577      0.0078
      0.0078      0.0877
PX =
      1.0000     -0.3875      0.6600
     -0.3875      1.0000      0.0104
      0.6600      0.0104      1.0000
Pxy =
      1.0000      0.1104
      0.1104      1.0000
```

【例 2】X, Y 中的每个列都作为独立变量的记录看待时，求 X, Y 的互协方差阵。

```
xbar=mean(X);[mx,nx]=size(X);
ybar=mean(Y);[my,ny]=size(Y);
CXY=(X-ones(mx,1)*xbar)'*(Y-ones(my,1)*ybar)/(mx-1)
xv=sqrt(diag(CX)),
yv=sqrt(diag(CY));
PXY=CXY./(xv*yv')
CXY =
     -0.0120      0.0001      -0.0039
      0.0206      0.0139       0.0066
     -0.0068     -0.0252       0.0148
PXY =
     -0.1809      0.0016      -0.0537
      0.3828      0.2576       0.1140
     -0.0782     -0.2867       0.1567
```

说明：在本例中求得的 CXY 与例 1 中的 Cxy 不同。

3.10.3 统计频数函数

对于随机样本$\{y_1, y_2, \cdots, y_n\}$构成的向量 Y，记 $y_{max} = \max\{Y\}$，$y_{min} = \min\{Y\}$。令 $L = y_{max} - y_{min}$，并把 L 分成 N 段，即 $l = \frac{L}{N}$；再把落在区间$(y_{min} + (i-1)l, y_{min} + il)$中的随机样本数记为 k_i；该区间的中心值记为 $x_i = (y_{min} + \frac{2i-1}{2}l)$。于是获得构成统计频数函数的两个统计向量 $K = [k_1, k_2, \cdots, k_N]$和 $X = [x_1, x_2, \cdots, x_N]$。

MATLAB 提供了两组实现这种计算的指令：

[K, X] = hist(Y, N)	在 N(缺省值为 10)个子区间上计算 Y 直方频数函数。
hist(Y, N)	用直方图表现在 N 个子区间上算得的 Y 频数函数。
[TK, RX] = rose(theta, N)	在 N(缺省值为 20)个子扇上计算 theta 的扇形频数函数。
rose(theta, N)	用扇形图表现在 N 个子扇上算得的 Y 频数函数。

【例 1】观察 10 000 个正态随机数的统计频数直方图和扇形图(图 3.10-1)。

```
randn('seed',10)
Y=randn(10000,1);
subplot(1,2,1)
hist(Y,30)
subplot(1,2,2)
rose(Y,30)
```

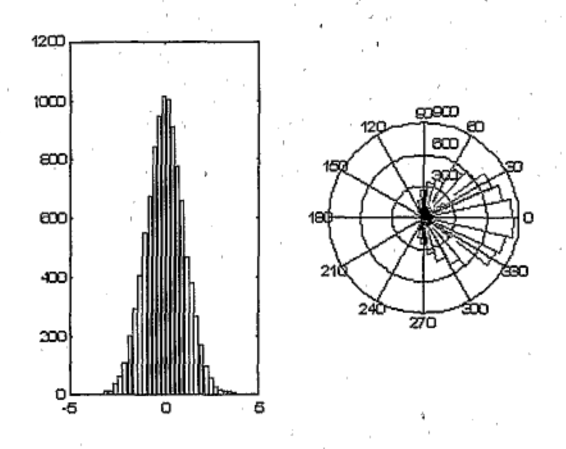


图 3 10-1　统计频数直方图和扇形图

3.10.4　有限差分和导数

DX = diff(X, k)　　按$(m\times n)$维 X 阵的列求$((m-k)\times n)$维 k（缺省值为 1）阶差分矩阵。
V = del2(U)　　求$(m\times n)$维 X 阵的$(m\times n)$五点格式差分矩阵。
dyx = diff(Y) / diff(X)　　Y, X 是同维矩阵，差分计算按列进行。
dyx = gradient(Y, dx)　　Y 是向量，dx（缺省值为 1）是 X 的分度值。
[GX, GY] = gridient(Z, dx, dy)　　GX、GY 分别是二元函数的$\frac{\partial Z}{\partial x}$、$\frac{\partial Z}{\partial y}$。
Cyx = gradient(Z, dx, dy)　　复数阵 Cxy 的实、虚部分别是$\frac{\partial Z}{\partial x}$、$\frac{\partial Z}{\partial y}$。

说明：

（1）五点格式差分算法源于拉普拉斯方程$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0$和泊松方程$\Delta^2 u=\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}$的差分解法。注意：MATLAB 4 2 以前版的用户指南和在线帮助上的有关解释与计算结果之间有差异（在边界上）。

（2）在求偏导数指令中，若输入宗量 dx，dy 取标量值，则它们表示求近似偏导数时自变量的分度值。若输入宗量 dx，dy 取向量值，则它们分别表示求近似偏导数的自变量绝对坐标值。dx 的长度与 Z 的列维相同，dy 的长度与 Z 的行维相同。

（3）在求偏导数指令中：GX 与 Z 同维，它给出在自变量绝对坐标值所决定网点上的$\frac{\partial Z}{\partial x}$偏导数值；GY 也与 Z 同维，它给出在自变量绝对坐标值所决定网点上的$\frac{\partial Z}{\partial y}$偏导数值。

【例 1】按列求差分。

```
X=[1,10,20;2,12,23;3,14,26;3,16,29]
D=diff(X)
X =
```

```
        1       10      20
        2       12      23
        3       14      26
        3       16      29
D =
        1        2       3
        1        2       3
        0        2       3
```

【例 2】五点格式差分。

```
[x,y]=meshgrid(-4:4,-3:3);
U=x.*x+y.*y
V4=4*del2(U)
U =
    25    18    13    10     9    10    13    18    25
    20    13     8     5     4     5     8    13    20
    17    10     5     2     1     2     5    10    17
    16     9     4     1     0     1     4     9    16
    17    10     5     2     1     2     5    10    17
    20    13     8     5     4     5     8    13    20
    25    18    13    10     9    10    13    18    25
V4 =
     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4
     4     4     4     4     4     4     4     4     4
```

说明：对于 $u=x^2+y^2, \Delta^2 u=4$。

【例 3】近似导数应用在三个不同位置上(图 3.10-2)。

```
x=0:pi/50:pi/2;
[m,n]=size(x);
y=sin(x);
dyx=diff(y)./diff(x);
x1=x(1,1:n-1);
x2=x1+pi/100;
x3=x1+pi/50;
```

```
plot(x1,dyx,'.r')
hold on
plot(x2,dyx,'-k')
plot(x3,dyx,'--b')
```

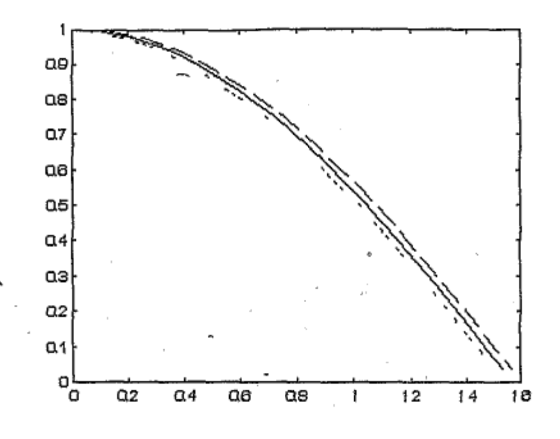


图 3 10-2 导数用于不同位置时的精度

说明.

(1) 所求得的导数 dyx 是(1×50)的向量,而 x 向量是(1×51)维。

(2) 图上最左边的导数曲线是利用横坐标向量 x1=[0,0 02*pi,…,0 48*pi] 画出的。

(3) 图上中间的导数曲线是利用横坐标向量 x2=[0 01*pi,0.03*pi,··,0.49*pi] 画出的。它最逼近真导数值。

(4) 图上最右边的导数曲线是利用横坐标向量 x3=[0 02*pi,0 04*pi,··,0 50*pi] 画出的。

【例 4】产生图 3 10-3 二元函数偏导数的几何表示的指令示例。

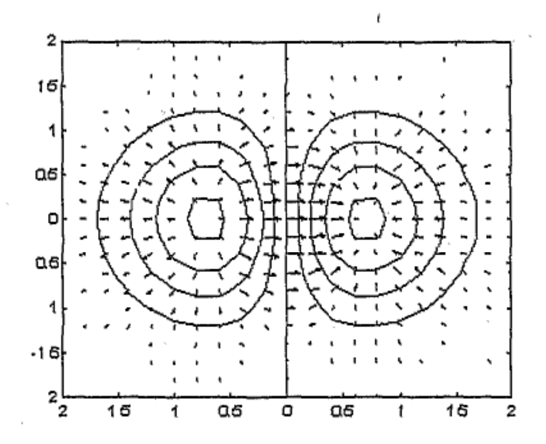


图 3 10-3 二元函数曲面的等高线和梯度

```
x=-2:0.2:2;y=-2:0 2:2;
```

```
[xx, yy] = meshgrid(x, y);
Z = xx  * exp( - xx ^2 - yy ^2);
[GX, GY] = gradient(Z,0 2,0 2);
colormap([1 0 0])
contour(x, y, Z, 'r'), , hold on, quiver(xx, yy, GX, GY, 'r'), hold off
```

说明:小箭头表示梯度,它是据 gradient 指令求得的偏导数画出的。

3.11 数字信号处理

向量可用来存放信号序列、采样数据,矩阵则可以按列存放"多通道"采样数据。MATLAB有专门的信号处理工具包(Signal Processing Toolbox)。目前版本已包含 80 多个函数指令。该信号处理工具包的内容十分丰富,已远远超出本节范围。本节仅对 MATLAB"主包"里若干信号处理指令作一些使用介绍。MATLAB"主包"有以下一些用于信号处理的基本指令:

abs	复数幅值
angle	复数相角
conv	卷积
con2	二维卷积
deconv	解卷
fft	快速傅里叶变换
fft2	二维快速傅里叶变换
filter	数字滤波器
filter2	二维数字滤波器
ifft	快速傅里叶逆变换
ifft2	二维快速傅里叶逆变换
fftshift	快速傅里叶结果的重整

3.11.1 快速傅里叶算法

FS = fft(X)	快速傅里叶变换。
FS = fft(X, n)	n 点快速傅里叶变换。
IFS = ifft(X)	快速傅里叶逆变换。
IFS = ifft(X, n)	n 点快速傅里叶逆变换。

说明:

(1) 对于矩阵 X,变换按列进行。

(2) 在 fft(X) 和 ifft(X) 指令中,当 X 的行维不能被 2 整除时,变换将采用速度较快的基-2 算法。否则,采用速度稍慢的混合基算法。

(3) 在 fft(X, n) 和 ifft(X, n) 指令中,当 X 的行维小于 n 时,用"0"补尾;反之,则截断。

【例 1】给定两个序列 x1，x2，其长度均为 5，利用 FFT 计算它们的线性卷积。

```
x1=[1 1 1 1 1];
x2=[2 1 3 1 1];
fx1=fft(x1,9)
fx2=fft(x2,9)
x=conv(x1,x2)
y=ifft(fx1.*fx2)
ry=real(y)
fx1 =
  Columns 1 through 4
    5.0000    0.5000 - 2.8356i   0.5000 + 0.1820i    0.5000 - 0.8660i
  Columns 5 through 8
    0.5000 + 0.4195i  0.5000 - 0.4195i  0.5000 + 0.8660i  0.5000 - 0.1820i
  Column 9
    0.5000 + 2.8356i
fx2 =
Columns 1 through 4
    8.0000      1.8473 - 4.8053i    -0.3794 - 0.5021i    0.5000 + 0.8660i
Columns 5 through 8
    3.0321 + 1.7051i    3.0321 - 1.7051i   0.5000 - 0.8660i    -0.3794 + 0.5021i
Column 9
1.8473 + 4.8053i
x =
     2     3     6     7     8     6     5     2     1
y =
Columns 1 through 4
   2.0000 - 0.0000i   3.0000 - 0.0000i   6.0000 - 0.0000i   7.0000 - 0.0000i
Columns 5 through 8
   8.0000 + 0.0000i   6.0000 + 0.0000i   5.0000 + 0.0000i   2.0000 + 0.0000i
Column 9
   1.0000 + 0.0000i
ry =
Columns 1 through 7
   2.0000   3.0000   6.0000   7.0000   8.0000   6.0000   5.0000
Columns 8 through 9
   2.0000   1.0000
```

说明：

(1) 两个序列的长度均为 5。为了能够用 FFT 进行线性卷积运算，应选择 $N \geqslant 5+5-1$，

分别进行 9 点的 FFT，然后再进行 9 点的 IFFT 。

(2) 快速傅里叶变换后的 fx1 , fx2 的第一个元素是直流分量，分别为向量 x1 和 x2 各元素之和。后 8 个元素是由 4 个共轭复数对组成。

(3) 由于计算截断误差，经逆变换后的数据会带很小的虚部，可用 real 指令将虚部删去。

(4) 以上结果表明，直接计算结果与通过变换算得的结果相同。

3.11.2 数据滤波

[Y, Zf] = filter(B, A, X, Zi)　　数字滤波。

说明：

(1) 该滤波器的数学模型为：

$$y_n = \frac{1}{a_1}(b_1x_n + b_2x_{n-1} + \cdots + b_{nb}x_{n-nb+1} - a_2y_{n-1} - \cdots - a_{na}y_{n-na+1})$$

在 filter 指令中，滤波器分母、分子向量为 $A=[a_1, a_2, \cdots, a_{na}]$、$B=[b_1, b_2, \cdots, b_{nb}]$；"移位寄存器"初始值为 $Z_i=[y_1, y_2, \cdots, y_{na-1}]$；滤波结束时，"移位寄存器"的存储值为 $Zf=[y_{N-na+1}, y_{N-na+2}, \cdots, y_N]$。被滤波数据向量为 $X=[x_1, x_2, \cdots, x_N]$。

(2) Z_i可以缺省。缺省时，认为"移位寄存器"初始值为全零。

(3) Z_f的主要用途是，对特别长的信号需要"切段"滤波时，使衔接不影响总的滤波质量。该宗量可以缺省。

【例 1】图 3 11-1 所示数字滤波过程的指令示范。

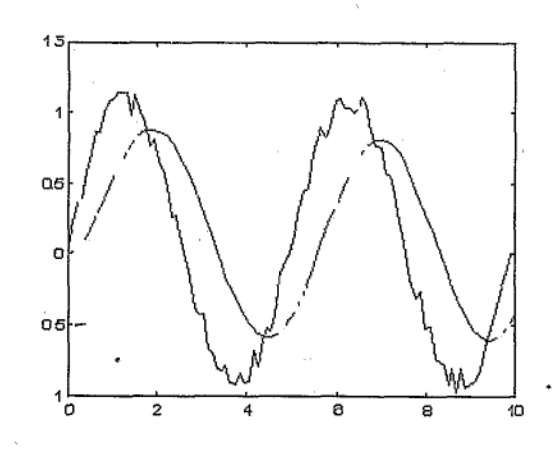


图 3 11-1　数字滤波器

```
t=linspace(0,10,100);          % 定义时间轴
s=sin(2*pi/5*t);               % 原始信号
noise=0.2*rand(size(t));       % 定义噪声
x=s+noise;                     % 带噪声的输入信号
y=zeros(size(x));
```

```
A=[1 -0.9];
B=[0.05 0.06];
y=filter(B,A,x);                    % 初始值为零的一阶 IIR 低通滤波器
plot(t,x,'b',t,y,'r')
```

3.12 稀疏矩阵

3.12.1 稀疏矩阵的存储方式

对于一个用矩阵描写的线性方程来说，n 个未知数的问题就涉及一个(n×n)的方程组。解这个方程就需要 n^2 个字的内存和正比于 n^3 的计算时间。这样，解几百阶、上千阶的方程就不那么容易处理。幸运的是，在大多数情况下，一个未知数只与数量不多的其他变量有关，即关系矩阵是稀疏的。

稀疏矩阵及其算法，就是不存储那些"0"元素，也不对它们进行操作，从而节省内存和时间开销。稀疏矩阵的计算复杂性和代价仅取决于稀疏矩阵的非零元素数目，而与该矩阵的总元素数目无关。

在 MATLAB 中有两种存储矩阵的方式：一种是全元素(Full)存储；另一种是稀疏(Sparse)存储。所谓稀疏存储，有以下特点：

(1) 只存储矩阵的非零元素，并且存储按列进行。

(2) 对于每列，用一个实数(或复数)数组记述非零元素值，再用一个整数数组记述相应非零元素的行下标。

实现两种存储方式转换的指令如下·

SM=sparse(A)　　把矩阵存储方式从任何一种形式(含全元素形式)转变为稀疏形式。

FM=full(A)　　把矩阵存储方式从任何一种形式(含稀疏形式)转变为全元素形式。

观察这两种存储方式的差别可以用 whos 指令实现(见第 3 12 3 节例)。

3.12.2 稀疏矩阵的创建

全元素矩阵经运算后仍然是全元素存储的。因此，假如不经过专门定义和运算，稀疏矩阵是不会自动产生的。但是，一旦某个矩阵以稀疏方式存储，那么由它参与运算的结果将仍以稀疏方式存储，除非运算本身(如多次加运算)使稀疏性消失。

稀疏矩阵创建指令：sparse

SM=sparse(I , J , S , m , n , nzmax)

说明：

(1) 用 [I , J , S] 的行创建(m×n)维稀疏矩阵 SM。

(2) S 是按列排列的所有非零元素构成的向量。I，J 分别是非零元素的行下标和列下标向量。这三个向量的长度相同。m，n 分别是待生成稀疏矩阵 SM 的行、列维。nzmax 是用来

为非零元素指定存储空间的正整数(它一定不小于非零元素总数)。

(3) 调用格式中的 6 个输入宗量在一定条件下可以缺省。详见用户指南或在线帮助。

稀疏带状矩阵创建指令:spdiags

SM = spsiags(B, d , m , n)

说明:

(1) m , n 分别指定矩阵 SM 的行、列维。d 是长度为 p 的整数向量,它指定 SM 的对角线位置;B 是全元素阵(但不必一定),用来给定 SM 的对角线位置上的元素,其行维为 min(m, n),列维为 p 。

(2) 该指令的其他调用格式请看用户指南或在线帮助。

(3) MATLAB 还提供了其他几种特殊稀疏矩阵的创建指令:稀疏单位阵(speye);稀疏随机阵(sprandn);稀疏对称随机阵(spandsym)。详见用户指南或在线帮助。

外部数据转换为稀疏矩阵的指令:spconvert

SM = spconvert(T)

说明:

(1) T 来自外部数据文件(可以是 .mat 文件,也可以是 .dat 文件)。数据文件由 load 指令装载于 MATLAB 内存空间。

(2) T 数组的行维为 nnz (即非零元素数)或 nnz + 1,(对实数而言)列维为 3 或(对复数而言)为 4 。T 数组的每一行(以 [I , J , Sre , Sim] 形式)指定一个稀疏矩阵元素。

(3) load , save 指令本身不区分矩阵的存储形式,都能正确操作。

【例 1】用两种不同方式创建三对角稀疏矩阵。

```
n=5;
SM1=sparse(1:n, 1:n, -2 * ones(1,n),n,n,n);
SM2=sparse(2:n, 1:n-1,ones(1,n-1),n,n,n-1);
S1=SM1+SM2+SM2'
e=ones(n,1);
S2=spdiags([e,-2*e,e],[-1,0,1],n,n)
SF=full(S1)
S1 =
   (1,1)      -2
   (2,1)       1
   (1,2)       1
   (2,2)      -2
   (3,2)       1
   (2,3)       1
   (3,3)      -2
   (4,3)       1
```

```
   (3,4)        1
   (4,4)       -2
   (5,4)        1
   (4,5)        1
   (5,5)       -2
S2 =
   (1,1)       -2
   (2,1)        1
   (1,2)        1
   (2,2)       -2
   (3,2)        1
   (2,3)        1
   (3,3)       -2
   (4,3)        1
   (3,4)        1
   (4,4)       -2
   (5,4)        1
   (4,5)        1
   (5,5)       -2
SF =
    -2     1     0     0     0
     1    -2     1     0     0
     0     1    -2     1     0
     0     0     1    -2     1
     0     0     0     1    -2
```

说明：

(1) 本例仅做演示，所以 n 取得很小。用户在计算机上可以试试 n 取较大值时的情况。

(2) 所得结果 S1、S2 的显示格式反映了稀疏矩阵的一般显示方式。读者可以把它们与全元素表达的 SF 作比较。

3.12.3 稀疏矩阵的运算

基本规则

总的说来，MATLAB 的各种指令适用于任何一种存储方式的矩阵。全元素矩阵运算总是以全元素形式给出结果。当有稀疏矩阵参与运算时，所得结果将遵循以下规则：

(1) 把矩阵变换到标量或定长向量的函数（如 size、nnz）总是给出全元素结果。

(2) 把标量或定长向量变换到矩阵的函数（如 diag、eye、ones、randn、zeros）总是给出全

元素结果；而能给出稀疏结果的相应指令有 spdiags、speye、sprandn、sparse。

(3) 从矩阵到矩阵或向量的"单矩阵"变换函数将以原矩阵(是稀疏，还是全元素)形式给出结果。如 chol(S)、diag(S)、max(S)、sum(S)、~S 等。

(4) 两个矩阵运算(如 +、*、\、|)操作后的结果一般是全元素形式，除非参与运算的两个矩阵都是稀疏的，或除非操作本身(如 .*、&)保留稀疏性。

(5) 参与的矩阵扩展(如 [A B ; C D])的子矩阵中，只要有一个是稀疏的，那么所得结果也是稀疏的。

(6) 在矩阵援引中，将仍以原矩阵(是稀疏，还是全元素)形式给出结果。若 S 阵稀疏而 Y 阵是全元素形式，不管 I、J 是标量还是向量，那么"右援引"Y=S(I, J)产生稀疏结果；而"左援引"Y(I, J)=S 产生全元素结果。

常用指令及应用举例

issparse	当矩阵稀疏时给出"1"，否则为"0"
nnz	矩阵的非零元素总数
nonzeros	矩阵的非零元素数值
nzmax	指定存放非零元素所需内存
spalloc	为非零元素配置内存
spfun	求各非零元素的函数值
spones	用 1 置换非零元素
colmmd	列最小度排序
colperm	计算列排序置换向量
dmperm	矩阵的 Dulmage-Mendelsohn 分解
randperm	随机置换向量
symmmd	对称最小度排序
symrcm	反向 Cuthill-McKee 排序
condest	估计范-1 条件数
normest	估计 2-范数
sprank	结构秩
gplot	依图论法则画"(无向)图"
spy	画稀疏结构图

说明：关于上述指令的详细解释请查用户指南或在线帮助。

【例 1】全元素矩阵、稀疏矩阵、最小排序稀疏矩阵三角分解所需时间的比较。

```
n=100;                          % 给出矩阵的阶数
A=sprandsym(n,0.03)+100*speye(n,n);
                                % 建立(100×100)随机正定稀疏矩阵
subplot(1,2,1)
spy(A,'b',10),
```

```
title('Spy plot of matrix A')
subplot(1,2,2)
d=symmmd(A);                      % 采用最小度排序算法
spy(A(d,d),'b',10),
title('Matrix A with Minimun degree ordering');
B=full(A);                        % 给出 A 的全元素形式
% 比较三个矩阵的 cholesky 三角分解的运算时间
format short e
tic, L1=chol(B); t1=toc       % 全元素时,cholesky 分解的计算时间
tic, L2=chol(A);t2=toc        % 稀疏时,cholesky 分解的计算时间
tic, L3=chol(A(d,d));t3=toc% 最小度排序时, cholesky 分解的计算时间
t1 =
  2.8000e-001
t2 =
  1.1000e-001
t3 =
  0.0000e-001
```

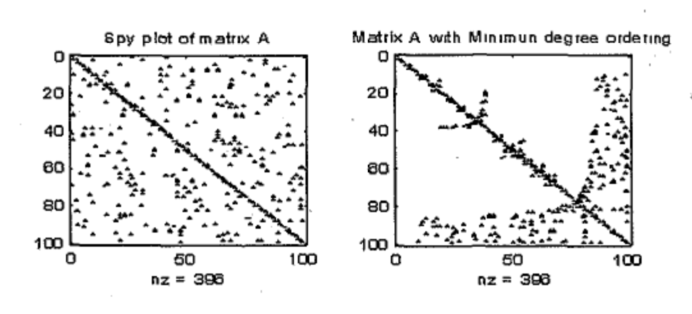


图 3 12-1 稀疏结构

说明:

(1) sprandsym(100,0 03) 创建密度为 0 03 的(100×100)正态随机对称稀疏矩阵,100 * speye(n,n) 则是保证矩阵 A 为主对角元占优的正定矩阵。

(2) 稀疏结构绘图指令 spy 中,第二宗量定义画图颜色,第三宗量使色点较大便于观察。

(3) 指令 symmmd 采用最小度排序算法。其中心思想是根据消元的每个阶段和选取可能的主元,获得填充项和运算次数的最小化,因而列高斯消元法能够与最小度算法较好地结合,该排序方法源于图论。

(4) 指令 full 将稀疏矩阵转化为全元素形式矩阵,用于比较。

(5) 计算结果表明,对全元素阵的 cholesky 分解所用时间最多,而经最小度排序后的稀疏阵分解所需用时最少。注意,计算用时随机器而变,本结果是在 486/DX-66 上的 Notebook 工作环境下所得。

(6) 用 whos 指令查看各变量的如下存储信息,进一步说明稀疏矩阵节省内存的作用。

Name	Size	Elements	Bytes	Density	Complex
A	100 by 100	396	5152	0.0396	No
B	100 by 100	10000	80000	Full	No
L1	100 by 100	10000	80000	Full	No
L2	100 by 100	1101	13612	0.1101	No
L3	100 by 100	492	6304	0.0492	No

说明：从这列表可以清楚看到矩阵维数、非零元素数、字节数、矩阵稀疏度、是否复数矩阵。

3.13 功能函数

前面介绍的 MATLAB 函数，几乎都是以数值矩阵或代表它的变量为输入宗量。在 MATLAB 中，还有两类函数：一类是以函数或代表它的函数名为输入宗量；另一类以符号表达式、符号矩阵或代表它们的变量为输入宗量。后一种函数涉及符号计算，它将在第四章介绍。本节着重介绍以函数为输入宗量的那种函数。它被称为功能函数(Function Functions)。这种函数主要用于：

(1) 对(输入宗量函数)求数值积分；

(2) 优化及求非线性方程的数值解；

(3) 求微分方程的数值解。

3.13.1 数值积分

求函数数值积分

S=quad('fname',a,b,tol,trace)　自适应递推辛普生(Simpson)法求数值积分。

S=quad8('fname',a,b,tol,trace) 自适应递推牛顿-柯西(Newton-Cotes)法求数值积分。

说明：

(1) 第一个输入宗量 'fmane' 是被积函数表达式字符串或函数文件名。

(2) 第二、三输入宗量 a 和 b 分别是积分的上、下限。

(3) 输入宗量 tol 用来控制积分精度。缺省时，默认 tol=0 001 。

(4) 最后输入宗量 trace，若取 1 则用图形展现积分过程，取 0 无图形。缺省时，不画图。

(5) quad8 比 quad 有更高的积分精度。但无论是 quad8 还是 quad，都不能处理可积的“软奇异点”。

【例 1】设 $y(t)=e^{-0\ 5t}\sin\left(t+\frac{\pi}{6}\right)dt$，求 $S=\int_0^{3\pi}y(t)dt$。

步骤一：用编辑器建立被积函数 esin.m 文件。

```
function y=esin(t)
y=exp(-0.5*t) *sin(t+pi/6);
```

说明：函数文件必须以“functıon”作为文件头。

步骤二：把写好的 esın m 文件存放于用户自己的工作目录（比方说，d：\ mydır）。

步骤三：在 MATLAB 环境下，利用下述命令使 d：\ mydır 成为当前工作目录。

```
cd d:\mydir
```

步骤四：调用积分指令。

```
S=quad('esin',0,3*pi)
```

（以下是所得结果）

```
S =
    0.9008
```

说明：在编制被积函数程序时，要注意所写程序应允许向量作为输入参量。正是出于这一考虑，在该函数文件中，采用了“数组乘”。

通过采样向量求数值积分

Z=trapz(X, Y)　　梯形法求数值积分。

Sc=cumsum(Y)　　欧拉法求积分函数值。

说明：

(1) 在 trapz 中，当 X，Y 是同维的列（行）向量时，所得的 Z 将是标量，它给出 Y 相对 X 的积分值；当 Y 是 $(m\times n)$ 维矩阵时，X 必须是 $(m\times 1)$ 的列向量，积分对 Y 各列分别进行，计算所得 $(1\times n)$ 维 Z 的元素是对应列函数对于 X 的积分。在该指令中，X 缺省时，认为 Y 为单位步长等距采样所得的函数阵。

(2) cumsum 的运算结果 Sc 与 Y 同维。Sc 的各列给出 Y 相应列的积分函数值，且认为积分采用的是等距单位步长。一般说来，该积分的精度较差。

【例 2】用梯形法和欧拉法求例 1 中的数值积分。

```
d=pi/1000;t=0:d:3*pi;nt=length(t);
Y=exp(-0 5*t) *sin(t+pi/6);
Z=trapz(Y)*d
Sc=cumsum(Y)*d;Scf=Sc(nt)
Z =
    0.9008
Scf =
    0.9016
```

3.13.2 优化和解非线性方程

本节介绍四个功能函数指令：求单变量函数最小值点指令 fmın；求多变量函数最小值点指令 fmıns；求单变量函数零点指令 fzero；求一般非线性方程组的解指令 fsolve 。前三个指令在 MATLAB“主包”里，最后一个在 MATLAB 优化工具包（Optımızatıon Toolbox）里。MATLAB 3 0 版中没有这些指令。

求函数的最小值点

[x, op] = fmin('fname', x1, x2, options, p1, p2, …, p10)

[x, op] = fmins('fname', x0, options, [ ], p1, p2, …, p10)

说明：

(1) fmin 用于求单变量函数最小值点。它有三个必不可少的输入宗量 'fname'、x1、x2。'fname' 是被最小化的目标函数名字字符串。输入宗量 x1, x2 限定自变量的取值范围，即 x1<x<x2 。

(2) fmins 采用著名的 Nelder－Mead 单纯形算法求解多变量函数的最小值点。'fname'、x0 是该指令两个不可缺省的输入宗量。x0 是最小值点的初始猜测值。

(3) options 是用来控制算法的参数向量。其中几个元素的意义是：

options(1)　　缺省值为 0 。若被赋予 1，则显示运算中间步骤。

options(2)　　缺省值为 0.0001 。它是中止寻优的容差值。

options(3)　　缺省值为 0.0001 。它是中止寻优的目标函数容差值。

options(14)　　缺省值为 500 。它是所允许的最大迭代的次数。

(4) p1, p2, …, p10 是向目标函数传送的参数，允许缺省。

(5) 第二个输出宗量 op 给出存放在 options(10) 中的实际运算迭代步数，允许缺省。

【例 1】当 $a=2$ 和 $a=8$ 时，求 $y(x)=x^3-ax-5$ 在 $0<x<5$ 中的最小值点。

步骤一：建立函数文件 f1.m。

```
function y=f1(x,a)
y=x.^3-a*x-5;
```

步骤二：用 fmin 指令求最小值点。

```
x2=fmin('f1',0,5,[],2)
x8=fmin('f1',0,5,[],8)
x2 =
    0.8165
x8 =
    1.6330
```

【例 2】著名的多维寻优测试函数——“香蕉”函数(Rosenbrock banana function)：$f(x)=100(x_2-x_1^2)^2+(a-x_1)^2$，在 $x_1=a, x_2=a^2$ 处有最小值。本例作数值寻优。

步骤一：建立“香蕉”函数文件 banana.m。

```
function f=banana(x,a)
f=100*(x(2)-x(1).^2).^2+(a-x(1)).^2;
```

步骤二：利用 fmins 寻优。

```
x=fmins('banana',[0,0],[0,1.e-8],[],sqrt(2))
x =
   1.4142      2.0000
```

说明:

(1) 在步骤一中,所建的"香蕉"函数以二元向量 x 为搜索变量。

(2) 在步骤二中,寻优从坐标原点开始;设中止寻优的自变量容差为(1 e-8);使 banana 函数中的 a 参数为($\sqrt{2}$)。

求单变量函数的零点

在第3 9 3 节中介绍了多项式的求根,而这里要介绍更一般的单变量函数的零点的求取指令。

z=fzero('fname', x0, tol, trace)

说明:

(1) 该指令能求比多项式(见 3 9 3 节)更一般的单变量函数的零点。

(2) 第一个输入宗量 'fmane' 是待求零点的函数文件名。x0 有二个作用:一是预定待搜索零点的大致位置;二是作为搜索起始点。'fmane' 和 x0 在指令调用中都不可缺少。

(3) 一个函数可能有多个零点,但本指令的结果 z 只给出离 x0 最近的那个零点。

(4) 输入宗量 tol 控制结果相对精度,可缺省,此时默认 tol=eps 。

(5) 最后一个输入宗量 trace 指定迭代信息是否在运算中显示,可缺省,此时默认 trace=0,不显示迭代信息。

【例 3】求函数 $y(t)=0\ 2t-e^{-0\ 5t}\sin\left(t+\frac{\pi}{6}\right)dt$ 在 t=2 附近的零点。

步骤一:建立函数文件 f2 m。

```
function y=f2(t)
y=0.2*t-exp(-0.5*t) *sin(t+pi/6);
```

步骤二:求零点。

```
z=fzero('f2',2)
z =
   1.6993
```

一般非线性方程组的数值解

这部分内容已超出 MATLAB 的"主包"范围,它将需要用到 MATLAB 的优化(Optimization)工具包。但由于非线性方程组是常遇到的一类数学问题,因此在本节中顺便作一简单介绍。对于一般非线性方程组 F(X)=0,其数值解 X 的求取指令如下:

X=fsolve('fun', X0)

说明:

(1) 函数文件 fun.m 用于定义所需求解的非线性方程组。

(2) X0 是对解 X 的初始猜测值。

【例 4】求 $\begin{cases}\sin(x)+y\hat{\ }2+\ln(z)=7\\3x+2\hat{\ }y-z\hat{\ }3+1=0\\x+y+z=5\end{cases}$ 的数值解。

步骤一:建立函数文件 myxyz m 。

```
function q=myxyz(p)
x=p(1);y=p(2);z=p(3);
q=zeros(3,1);
q(1)=sin(x)+y^2+log(z)-7;
q(2)=3*x+2^y-z^3+1;
q(3)=x+y+z-5;
```

步骤二：在 MATLAB 中运行以下指令。

```
xyz0=[1,1,1];
xyz=fsolve('myxyz',xyz0)
xyz =
      0.5990      2.3959      2.0050
```

说明：

(1) 该指令的调用格式默认：所求解的精度为 0 0001；所解方程成立精度为 0 0001。

(2) MATLAB 3 0 版没有该指令。

(3) 从原理上讲，该非线性方程组也可用(MATLAB 4 0 以上版本有的)符号工具包中的 solve 及 vpa 指令求数值解。但据笔者实践，这种计算方法所需时间是用户难以接受的。

3.13.3 微分方程的数值解

微分方程的数值解

[t, x] = ode23('xprime', t0, tf, x0, tol, trace)

[t, x] = ode45('xprime', t0, tf, x0, tol, trace)

说明：

(1) 两个指令的调用格式完全相同，均采用 Runge－Kutta 法。

(2) 该指令是对一阶常微分方程组设计的。因此，假如待解的是高阶微分方程，那么它必须先被演化为形如 x = f(x, t)的一阶微分方程组，即"状态方程"。

(3) 第一个输入参量 'xprime' 是定义 f(x, t)的函数文件名。该函数文件必须以 x 为输出；以 t, x 为输入宗量(注意变量的次序不可颠倒)。

(4) 输入宗量 t0 和 tf，分别是积分的起始值和终止值。

(5) 输入宗量 x0 是初始状态列向量。

(6) 输出宗量 t 和 x，分别给出"时间"向量和相应的状态向量。

(7) 第五个输入宗量 tol 控制解的精度，可缺省。缺省时，在 ode23 中默认，tol = 1.e－3；在 ode45 中默认，tol = 1 e－6 。

(8) 第六个输入宗量 trace 决定求解的中间结果是否显示，可缺省。缺省时，默认 trace = 0，不显示中间结果。

(9) 一般说来，ode45 比 ode23 的积分分段少，而运算速度更快。

【例 1】求著名的 Van der Pol 方程 $x+(x^2-1)x+x=0$。

步骤一：演化为状态方程。

令 $x_1=\dot{x}, x_2=x$，那么可把 $x+(x^2-1)\dot{x}+x=0$ 写成状态方程形式 $\begin{cases} x_1=(1-x_2^2)x_1-x_2 \\ x_2=x_1 \end{cases}$

步骤二：建立函数文件 xprime m。

```
function xdot=xprime(t,x)
xdot=zeros(2,1);
xdot(1)=(1-x(2)^2)*x(1)-x(2);
xdot(2)=x(1);
```

说明：

(1) 注意函数文件第一行：函数的输入参量一定是两个；次序一定是先“时间变量”，后“状态变量”。

(2) 函数文件的第二行使 xdot 成为二元零向量(建议采用列向量，以便被 MATLAB 其他指令调用)，这行是为后两行对 xdot 元素的调用作准备。

步骤三：解微分方程。

```
t0=0;tf=20;
x0=[0,0.25]';
[t,x]=ode23('xprime',t0,tf,x0);
plot(t,x(:,1),':b',t,x(:,2),'-r')
legend('速度','位移')
```

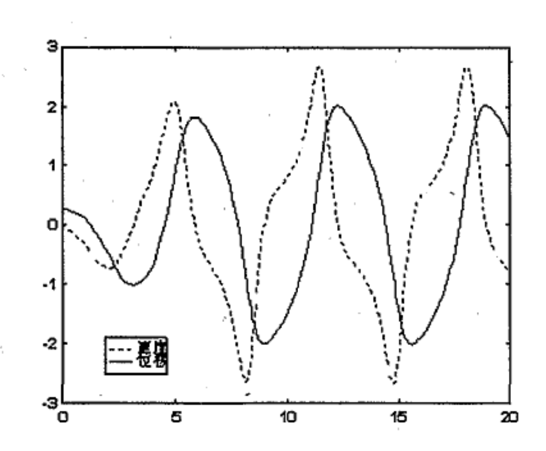


图 3 13-1　Van der Pol 方程的解

动态状态轨迹图示

ode23p('xprime',t0,tf,x0,tol)

说明：

(1) 该指令(MATLAB 4.0 以上版有此指令)直接给出状态方程状态轨迹。

(2) 该指令不给数值输出，而只给前三个状态的动态轨迹。

(3) 该指令的输人宗量的含义与 ode23 相同。

(4) 在使用该指令前，最好先定义状态空间的轴的范围。

【例 2】利用上例的函数文件 xprime.m 和初始条件，图示 Van der Pol 方程的相轨迹。

```
clf;
axis([-3,3,-3,3]);hold on
ode23p('xprime',t0,tf,x0)
```

说明：由于现有 MATLAB 的 Notebook 不具备表现动态图形的能力，因此请用户在 MATLAB的指令窗(MATLAB Command window)中运行上述指令，并在它的图形窗(Figure)中观察 ode23p 给出的动态轨迹。

第四章　MATLAB 的符号计算功能

除了数值计算以外，在数学、物理、应用科学和工程中还经常遇到符号计算问题。MATLAB开发和销售商 MathWorks 公司于 1993 年从加拿大滑铁卢大学购得了对 MAPLE 的使用权。随后，MathWorks 公司以 MAPLE 的“内核”为符号计算“引擎(Engine)”，依赖 MAPLE 已有的库(Library)，开发了在 MATLAB 环境下实现符号计算的工具包 Symbolic Math Toolbox。

Symbolic Math Toolbox 第一版(1994 年 1 月推出)是以 MAPLE V2 为基础开发的，约 10M 字节。它可在 MATLAB 4.0，4.1，4.2 等版本下使用，但不在 MATLAB“主包”内，需要额外安装。安装过程不再讲述。读者可以参考关于 MATLAB 主程序的安装方法或 MATLAB 用户指南。

Symbolic Math Toolbox 第一版与 MAPLE 交换信息有下面三个通道：

(1) 通过调用由 MATLAB 语言写成的 60 多个专用函数文件进行符号运算。对熟悉 MATLAB 工作方式的用户来说，这种使用方式比较容易掌握。专用符号计算函数文件的内容包括：符号表达式与矩阵的操作；微积分；线性代数；方程的求解；简约与展开；特殊数学函数。有关这部分内容将分九节(从第 4.2 节到第 4.9 节)作较详细的介绍。

(2) 通过两个专门设计的 M 文件(maple.m，mpa.m)进行符号计算。当用这种方式进行符号计算时，用户需要掌握 MAPLE 的一些基本语言。凡是 MATLAB 没有提供专用符号计算函数的其他符号计算可通过这 maple.m 文件进行。具体使用方法和注意事项将在第 4.10 节中叙述。

(3) 通过调用函数计算器(Function Calculator)来进行符号计算。这是最方便最直观的符号计算方法。但函数计算器的功能比较简单，它只能进行不多于两个函数的代数及微积分符号计算。第 4.11 节将专门讲述计算器的使用。

在 Symbolic Math Toolbox 中还有在 MATLAB 环境下找到符号计算指令使用方法的在线求助功能。这将在第 4.12 节讲述。

假如读者想较快地领略符号计算的魅力，那么请先实践一下第 4.1 节中的例题。在此要提请读者注意：符号计算的整个过程是以字符进行的，即便是那些以数字形式出现的量也是字符量，而不是数值量。

4.1　入　门

4.1.1　符号计算入门

符号计算的特点之一是：运算对象和过程允许存在非数值的符号变量。本节将以一些简单的实例让读者初步体验，符号计算是什么？MATLAB 符号数学工具包是怎么工作的？

【例 1】求函数 $f(x)=\sin^2(x)$ 的微积分。

```
f='sin(x)^2';                    %定义函数f(x)
dfdx=symdiff(f)                  %求d/dx f(x)的指令
intf=int(f)                      %求∫f(x)dx的指令
```

（下面是显示的计算结果内容）

```
dfdx =
2 * sin(x) * cos(x)
intf =
-1/2 * sin(x) * cos(x) + 1/2 * x
```

【例 2】求 f(x)在 x=0 处的泰勒级数，最后一项为 8 阶无穷小量符号。

```
Tf=taylor(f,8)                   % 求泰勒级数
Tf =
1 * x^2 - 1/3 * x^4 + 2/45 * x^6 + O(x^8)
```

【例 3】求矩阵 $A=\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ 的行列式值、逆、特征根。

```
A=sym('[a11,a12;a21,a22]')       % 定义矩阵A的指令
DA=determ(A)                     % 求矩阵A行列式值的指令
IA=inverse(A)                    % 求矩阵A的逆
EA=eigensys(A)                   % 求矩阵A特征根的指令
A =
[a11,a12]
[a21,a22]
DA =
a11 * a22 - a12 * a21
IA =
[-a22/(-a11 * a22 + a12 * a21), a12/(-a11 * a22 + a12 * a21)]
[ a21/(-a11 * a22 + a12 * a21), -a11/(-a11 * a22 + a12 * a21)]
EA =
[1/2 * a11 + 1/2 * a22 + 1/2 * (a11^2 - 2 * a11 * a22 + a22^2 + 4 * a12 * a21)^(1/2)]
[1/2 * a11 + 1/2 * a22 - 1/2 * (a11^2 - 2 * a11 * a22 + a22^2 + 4 * a12 * a21)^(1/2)]
```

4.1.2　任意精度计算入门

符号计算的另一个特点是，可以获得任意精度的数值解。先看以下算例。

【例 1】求方程 $x^2-x-1=0$ 的精确解和任意精度解。

```
R1=solve('x^2-x-1=0')          % 求方程的精确解的指令
RV=vpa(R1)                     % 以默认精度求近似解
RV4=vpa(R1,4)                  % 求 4 位数字有效的近似解
RV20=vpa(R1,20)                % 求 20 位有效数字的近似解
R1 =
[1/2*5^(1/2)+1/2]
[1/2-1/2*5^(1/2)]
RV =
[ 1.618033988749895]
[-.6180339887498950]
RV4 =
[ 1.618]
[-.6180]
RV20 =
[ 1.6180339887498948482]
[-.61803398874989484820]
```

在例 1 中，方程解的四种表达形式都是由符号计算获得的。即使它们形式上是数值，但从变量分类上看，它们仍是字符串。至于怎样从精确解获得任意精度解，怎样改变默认精度，怎样把任意精度符号解变成“真正”数值解，则需要运用以下指令实现。

digits(n)	使该指令后的近似解的精度为 n 位有效数字。
subs(S, Dsym, Fsym)	把 S 符号解中的自由参数 Fsym 用数字字符 Dsym 代替。
vpa(S, n)	求 S 的 n 位有效数字近似解。n 缺省时，给出默认精度近似解。
numeric(S)	把不含自由参数的 S 符号变量转化为数值变量。

【例 2】4 1 1 节例 1 中，对于函数 $f(x)=\sin^2(x)$，求解 $\frac{d}{dx}f(x)$ 各种表现形式的转换。

```
Ddfdx=subs(dfdx,'pi/3','x')
digits(30)
Vdfdx=vpa(Ddfdx)
Ndfdx=numeric(Ddfdx)
Ddfdx =
2*sin(1/3*pi)*cos(1/3*pi)
Vdfdx =
.866025403784438646763723170755
Ndfdx =
    0.86602540378444
```

说明：

(1) 指令 digits(30) 把符号近似解的默认 16 位精度设置为 30 位近似精度。

(2) 假若用 isstr 指令检查以上三个变量，可以验证 Ddfdx、Vdfdx 是符号变量，而 Ndfdx 是数值变量。

4.2 符号表达式和符号矩阵的创建

在数值计算(包括输入、输出及中间计算在内的)过程中，所运作的变量都是被赋了值的数值变量。而在符号计算的整个过程中，所运作的是符号变量(Symbolic variable)。在符号计算中所出现的数字也都是当作符号处理的。

4.2.1 符号表达式和符号方程的创建

符号表达式(Symbolic expression)和符号方程(Symbolic equation)是两种不同的操作对象。它们的区别在于：前者不包含等号(=)，而后者必须带等号。但这两种对象的创建方式是相同的。它们最简单和最常用的创建方式与 MATLAB 中的字符串变量的创建几乎相同。例如：

```
f='sin(x)^2';                % 所创建的函数 sin^2(x)赋给变量 f
eq='a*x^2+b*x+c=0';          % 所创建的方程 ax^2+bx+c=0 赋给变量 eq
de='Dy+y^2=1';               % 所创建的微分方程 y'+y^2=1 赋给变量 de
```

在此要提请注意的是：

(1) 第一个等号右边的部分，既可以是符号表达式，也可以是符号方程。

(2) 以上三个例子都把创建的符号表达式或符号方程赋给了符号变量。引入符号变量的目的是为以后调用方便，但这并不是必需的。事实上，符号表达式可以直接参与运算。

(3) 符号表达式和符号方程对空格都是敏感的。因此，在创建符号表达式时，不要在字符间任意乱加“修饰性”空格符。由于符号表达式可以看作(1×1)的符号矩阵，因此它也可以用 sym 指令创建。如 f=sym('sin(x)^2') 与 f='sin(x)^2' 的作用完全一样。

4.2.2 符号矩阵的创建和修改

这小节着重叙述 sym 指令在创建、援引和修改符号矩阵中的使用方法。

sym 指令创建符号矩阵的直接输入法

这是模仿 MATLAB 数值矩阵的直接输入法设计的。矩阵元素可以是任何(不带等号的)符号表达式；各矩阵元素的长度可以不同；矩阵行之间用分号(;)隔断；各矩阵元素间用逗号(,)分隔。例如，在 MATLAB 环境下运行

```
msy=sym('[1/(a+x),sin(x),(b-x)/(a+x);1,exp(x),x^2]')
```

就显示结果

```
msy =
[1/(a+x),sin(x),(b-x)/(a+x)]
```

```
[1,exp(x),                    x^2]
```

创建符号矩阵的字符串直接输入法

这是模仿 MATLAB 字符串矩阵的直接输入法设计的。在这种方法中，不需要调用 sym 指令，但要保证同一列中各元素字符串有同样的长度。为此，在较短字符串的前后可用空格符填充。例如，上述的 msy 矩阵也可以用以下指令创建

```
msy=['[1/(a+x),sin(x),(b-x)/(a+x)]';'[1       ,exp(x),      x^2      ]'];
```

注意：符号矩阵的每一行(row)的两端都有方括号[ ]。这是与 MATLAB 字符串矩阵的一个重要区别。

把数值矩阵转化为符号矩阵

MATLAB 中的数值矩阵不能直接参与符号运算，必须通过转化才行。指令 sym(Mnu)就能把数值矩阵 Mnu 转化为符号矩阵。不管数值矩阵 Mnu 的元素原先是用分数还是浮点数表达，转化后的符号矩阵都将以最接近的精确有理形式给出。例如有数值矩阵

```
Mnu=[2/3,sqrt(3)/3,0.333;2.5,1/0.7,log(3)]
Mnu =
    0.6667    0.5774    0.3330
    2.5000    1.4286    1.0986
```

可用以下指令将把 Mnu 转化为符号矩阵

```
sym(Mnu)
ans =
[2/3, 1300077228592327*2^(-51), 1499698675914375*2^(-52)]
[5/2,                     10/7, 2473854946935174*2^(-51)]
```

把这个指令与 MATLAB 中许多数值特殊矩阵生成指令(如 eye, ones, zeros, magic)配合使用，可以产生许多特殊的符号矩阵。

利用元素通式创建符号矩阵

这种创建方式下的一般调用格式是

```
sym(m,n,'expression(i,j,x,y,…)')
sym(m,n,'r','c','expression(r,c,x,y,…)')
```

其中，m 是符号矩阵的行维数，n 是符号矩阵的列维数，i、j 分别是元素通式中默认的行、列序号参量，其取值范围为 i=1:m, j=1:n，r、c 分别是元素通式中的行、列序号参量，其取值范围为 r=1:m, c=1:n，expression(…,x,y,…)是元素通式，可以是符号表达式，它包含 x，y 等自由变量。

【例 1】sym 指令使用。

```
Msy1=sym(2,3,'r','c','1/(r+c+s)')        % 创建(2×3)符号矩阵指令
Msy1 =
```

```
[1/(2+s), 1/(3+s), 1/(4+s)]
[1/(3+s), 1/(4+s), 1/(5+s)]
```

说明：上述创建指令中的'r'和'c'分别是元素通式(General Expression)中的行、列序号变量。当行、列序号变量分别用 i、j 表示时，上述符号矩阵 Msy1 的创建指令可更简洁地写成 Msy1=sym(2,3,'1/(i+j+s)')。

符号矩阵元素的援引和修改

在数值计算中，通过一个指令就可实现对矩阵任何一个子阵的援引和修改。但在符号计算中，援引(Quote)和修改(Modify)只能对符号矩阵的元素一个一个地进行。

sym(S,i,j)　　援引指令：将给出符号矩阵 S 的第(i,j)个元素。
sym(S,i,j,'expr')　　修改指令：将用 expr 替代符号矩阵 S 中原先的第(i,j)个元素。

【例 2】修改矩阵元素。

```
Msy1=sym(Msy1,1,1,'0')          % 修改矩阵 Msy1 的第(1,1)号元素为零
Msy1 =
[        0,   1/(3+s),   1/(4+s)]
[ 1/(3+s),    1/(4+s),   1/(5+s)]      % 修改后的结果
```

4.3 符号矩阵的基本运算

在 MATLAB 的数值计算中，矩阵的加、减、乘、除等运算的操作指令都很直观简单。在符号计算中，情况就不同了，所有涉及符号计算的操作都要借助专用函数进行。由于 Symbolic math toolbox 第一版是在 MATLAB 和 MAPLE 各自的计算“引擎”都没作任何修改的情况下设计的，所以符号计算指令与数值计算指令的这种差异就不可避免。

以下对符号矩阵适用的指令，都适用于符号表达式。当然，这些指令对符号方程是没有意义的。

4.3.1 符号矩阵的加、减、乘运算

实现符号矩阵加(Add)、减(Subtract)、乘(Multiply)的指令分别是

symadd(A,B)　　给出两个符号矩阵的和 A+B。
symsub(A,B)　　给出两个符号矩阵的差 A−B。
symmul(A,B)　　给出乘积 A×B。

说明：

(1) symadd(A, −B) 决不意味着 A+(−B)=A−B，因为(−B)对符号运算是非法的。

(2) 相乘规则是：符号表达式可以与符号矩阵相乘；两个内维数相同的符号矩阵可以相乘。

【例 1】$\frac{1}{(s+1)(s+3)}$与$\begin{bmatrix}(s+1) & s \\ & (s+1\ 5)\end{bmatrix}$的相乘,并把结果赋给 G。

```
G=symmul('1/(s+1)/(s+3)',sym('[s+1,s;0,s+1.5]'))
G =
[1/(s+3),          1/(s+1)/(s+3)*s]
[         0, 1/(s+1)/(s+3)*(s+1.5)]
```

【例 2】矩阵$\begin{bmatrix}(s+1) & s \\ & (s+1\ 5)\end{bmatrix}$与向量$\begin{bmatrix}\frac{1}{s(s+2)} & \frac{1}{(s+1)(s+2)}\end{bmatrix}^T$ 相乘。

```
symmul(sym('[s+1,s;0,s+1.5]'),sym('[1/(s*(s+2));1/((s+1)*(s+2))]'))
ans =
[        (2*s^2+2*s+1)/s/(s+2)/(s+1)]
[(s+1.500000000000000)/(s+2.)/(s+1.)]
```

4.3.2 符号矩阵的逆和除运算

符号矩阵的求逆(Inverse)指令和符号矩阵的除(Divide)运算指令如下:

inverse(B)　　　　求 B^{-1}。

symdiv(A,B)　　　实现 A/B,即 AB^{-1}。

说明:

(1) A 可以是符号表达式或(m×n)维符号矩阵;B 可以是符号表达式或(n×n)维满秩符号阵。

(2) inverse(B) 与 symdiv('1',B) 等价,而 symmul(A, inverse(B)) 与 symdiv(A,B) 等价。

4.3.3 符号矩阵的幂运算

与数值计算相比,符号计算对幂(Power)运算 S^p 所加的限制条件更多,否则所得结果就很难保证正确。具体如下:

sympow(S,p)　　　求幂运算指令 S^p

说明:若 S 为标量符号表达式,p 可为标量符号或数值表达式;若 S 为符号方阵,p 必须为整数。

4.3.4 符号矩阵的综合运算指令

除了前面几小节介绍的加、减、乘、除、幂等单种符号计算指令外,在符号数学工具包中还有一个综合运算指令 symop 。提供此指令是出于两个考虑:(1)实际使用的需要。当多个符号矩阵或混有数值矩阵,要实现相互间的多种运算时,假如仅依靠单种运算指令去做,就显得

缺乏效率;(2)软件编制上的方便。实际上,那些单种运算文件,都是在综合运算函数文件 symop m 基础上开发的。

符号矩阵的综合运算(Oprations)指令如下:

symop(s1 ,s2,s3,…)　　符号矩阵的综合运算。

　　s1,s2,s3,…　　分别是符号矩阵或数值矩阵或 '+','-','*','/','^','(',')' 中某算符。

【例1】综合运算指令与多次调用单种运算指令的比较。

```
A=sym('[a1,a2;a3,a4]');B=sym('[b1,b2;0,b4]');N=[1,2;3,4];
F=symop(A,'/','(',B,'+',N,')')
F =
[(a1*b4+4*a1-3*a2)/(b1*b4+4*b1+b4-2-3*b2), -(a1*b2+2*a1-a2
*b1-a2)/(b1*b4+4*b1+b4-2-3*b2)]
[(a3*b4+4*a3-3*a4)/(b1*b4+4*b1+b4-2-3*b2), -(a3*b2+2*a3-a4
*b1-a4)/(b1*b4+4*b1+b4-2-3*b2)]
FF=symdiv(A,symadd(B,sym(N)))          % 由单种运算指令复合而成的输入指令
FF =
[(a1*b4+4*a1-3*a2)/(b1*b4+4*b1+b4-2-3*b2), -(a1*b2+2*a1-a2
*b1-a2)/(b1*b4+4*b1+b4-2-3*b2)]
[(a3*b4+4*a3-3*a4)/(b1*b4+4*b1+b4-2-3*b2), -(a3*b2+2*a3-a4
*b1-a4)/(b1*b4+4*b1+b4-2-3*b2)]
```

说明:

(1) 当综合运算指令中同时存在符号表达式和符号矩阵时,所得结果不一定正确。

(2) 当综合运算指令中括号“(…)”两侧同时存在乘、除运算时,所得结果不一定正确。

鉴于综合运算函数 symop m 文件设计中的缺陷,本书作者强烈建议:用户应优先采用单种运算指令的复合调用去完成比较复杂的符号矩阵运算。

4.4 因式分解、展开和简化

4.4.1 因式分解、展开

对符号表达式、符号矩阵可进行如下操作。

factor(S)　　对S进行因式分解。S可以是单个表达式、有理分式、方程、矩阵。

expand(S)　　对符号矩阵S进行展开。

collect(S)　　把S符号矩阵元素中“x”的同幂项系数进行合并。

collect(S,v)　　把S符号矩阵元素中“v”的同幂项系数进行合并。

【例 1】符号矩阵的分解。

```
G=sym('[(s+1)/(s^2+5*s+6),2*(s+1)/(s^2+2*s-3);(s^2-s)/(s^2+2*s+1),s/(s+2)]');
factor(G)
ans =
[(s+1)/(s+3)/(s+2), 2*(s+1)/(s+3)/(s-1)]
[   s*(s-1)/(s+1)^2,              s/(s+2)]
```

【例 2】符号矩阵的展开。

```
GS=sym('[sin(x+y)/cos(2*x);exp(x-y);log(a*b^2/c);(s+1)*(s+3)/((s+2)*(s+4))]');
expand(GS)
ans =
[1/(2*cos(x)^2-1)*sin(x)*cos(y)+1/(2*cos(x)^2-1)*cos(x)*sin(y)]
[                                              exp(x)/exp(y)]
[                                    log(a)+2*log(b)-log(c)]
[           1/(s+2)/(s+4)*s^2+4/(s+2)/(s+4)*s+3/(s+2)/(s+4)]
```

说明:展开将对该矩阵元素中的因式乘积、三角函数、指数函数、对数函数进行,但在分母的因式乘积将不展开。

【例 3】同幂项系数的合并。

```
S='a*sin(t)*x^2+b/t*sin(t)*x^2+cos(t)*x+sin(t)^2*x+log(b)*sin(t)^2';
XS=collect(S)                % 合并"x"的同幂项系数
SS=collect(S,'sin(t)')       % 合并"sin(t)"的同幂项系数
XS =
(b/t*sin(t)+a*sin(t))*x^2+(cos(t)+sin(t)^2)*x+log(b)*sin(t)^2
SS =
(x+log(b))*sin(t)^2+(a*x^2+b/t*x^2)*sin(t)+cos(t)*x
```

4.4.2 符号矩阵的简化

本小节介绍综合性简化函数 simple(S),该指令将使符号矩阵 S 中的每个元素以最简短的形式给出。为了实现这一目的,该 simple.m 文件依次从 MAPLE 中调用了如下功能:

(1) simplify 利用函数规则(如三角函数恒等式)对表达式进行简化。

(2) radsimp 对包含根式的表达式进行简化。

(3) combine 把表达式中以求和形式、乘积形式、幂次形式的各项(Terms)归并成单项。在许多情况下,该变换是 expand 的反变换。

(4) factor 实现多变量多项式的因式分解。

(5) convert 把表达式从一种形式转化成另一种形式。比如,它可实现从多项式到嵌套式(Horner 或 Nested)的转化。

(6) collect 合并同幂项系数。

综合性简化函数有以下两种调用格式:

simple(S)	显示 S 被简化的中间过程,并给出最简结果。
[R,HOW]=simple(S)	不显示简化过程。R 为简化结果,HOW 为特别的简化操作。

【例 1】符号向量的简化。

```
S=sym('[sqrt(x^2+2*x+1); sin(x)^2+cos(x)^2;(a+b*i)*(a-b*i)]');
[R,how]=simple(S)
R =
[     1+x]
[        1]
[a^2+b^2]
how =
combine(trig)
```

说明:最后一个显示结果表明,简化过程中用了三角函数简化工具。

4.4.3 符号变量替换

在第 4 1 2 节中,已经介绍了如何利用变量替换指令把符号解中的自由参数替换成数字符。本节要更全面地介绍变量替换指令。实现这功能的指令:一个是 subs ,它是利用 MAPLE 中的 subs 指令编的,适用于单个符号矩阵、符号表达式、符号代数和微分方程;另一个是 symvars ,它由 MATLAB 原有的基本指令构成,适用于符号表达式组、符号代数方程组,替换速度快。它们的具体使用格式如下。

subs(S, NEW)	用新变量 NEW 替代 S 中的默认变量。
subs(S, NEW, OLD)	用新变量 NEW 替代 S 中的指定变量 OLD。
symvars(S, NEW, OLD)	用新变量 NEW 替代 S 中的指定变量 OLD。

【例 1】符号矩阵的变量替换。

```
G=sym('[a*sin(b*x),a+b;exp(a*x),sqrt(x)]');
G1=subs(G,'pi/3')          % 用"pi/3"替换 G 中默认变量"x"指令
G2=subs(G,'2','a')         % 用"2"替代 G 中的变量"a"指令
G1 =
[a*sin(1/3*b*pi),                   a+b]
[   exp(1/3*a*pi),   1/3*3^(1/2)*pi^(1/2)]
G2 =
[2*sin(b*x),        2+b]
```

```
[    exp(2 * x),     x^(1/2)]
```

【例 2】求两个复数的乘积。

```
c='c1+c2 * i';b='b1+b2 * i';
cb=symvars(collect(symvars(expand(symmul(c,b)),'x','i')),'i','x')
                                        % symmul 实现符号乘
                                        % expand 把所乘结果展开
                                        % symvars 把展开式中的"i"替换成"x"
                                        % collect 把展开式中的同类项合并
                                        % symvars 把合并后式中的"x"替换回"i"
cb =
(b2 * c1 + b1 * c2) * i + c1 * b1 - c2 * b2
```

说明：在符号矩阵中，小写字母"i"被默认为"虚数单元(Imaginary unit)"。

4.4.4 确定符号变量

在具体介绍指令使用格式之前，需要先解释一下该符号数学工具包确定符号变量的规则：(1)只对(除"i""j"以外的)单个小写英文字母进行搜索。(2)小写字母"x"是首选符号变量。(3)其余小写字母被选中为符号变量的次序是，在英文字母表中，靠"x"近的优先，在"x"之后的优先。

在以下 symvar 指令的使用格式中，S 可以是任何符号对象(表达式或矩阵)。

symvar(S)　　　　确定单变量符号对象 S 中的符号变量。

symvar(S,N)　　　在 N 指定数量范围内搜索符号对象 S 中的变量。

说明：

(1) N 可以是正整数，也可以是指定变量个数范围的二元行向量。

(2) 当 S 中(除"i""j"以外)单个小写英文字母数不等于 N 或小于 min(N)时，给出"出错"警告信息。

(3) 当 S 中(除"i""j"以外)单个小写英文字母数大于 max(N)时，给出"NaN"。

4.5 符号矩阵分解

与数值计算一样，在符号计算中，符号矩阵分解也特别有用。计算符号矩阵维数、转置、行列式、特性值分解、奇异值分解、零空间和列空间分解的指令如下：

symsize(S)　　　　求 S 矩阵维数指令。

transpose(S)　　　求 S 的符号转置矩阵。

determ(S)　　　　求 S 阵的行列式。

colspace(S)　　　给出 S 列空间的基。

symsize(colspace(S),2)　给出 S 的秩。

nullspace(S)　　　　　给出 S 零空间的基。
symsize(nullspace(S),2)　给出 S 的零化度。
[VE,E]=eigensys(S)　　VE 给出 S 的特征向量,E 给出阵 S 的特征值。
[VJ,J]=jordan(S)　　　VJ 给出 S 的广义特征向量,J 给出相应的约当标准形。
singvals(A)　　　　　给出一般符号阵 A 的奇异值。
[U,S,V]=singvals(A)　给出(不带自由变量的)A 阵的奇异分解三对组。

【例 1】无重根阵的分解。

```
D=sym('[1,-3;2,2/3]');
[VE,E]=eigensys(D)
VE =
[1/2*E(1)-1/3, 1/2*E(2)-1/3]
[            1,            1]
E =
[5/6+1/6*i*215^(1/2)]
[5/6-1/6*i*215^(1/2)]
```

说明:在 VE 特征向量阵中的 E(1)、E(2)分别是特征根向量 E 中第一元和第二元。

【例 2】有重根阵的分解。

```
C=sym('[1,1,2;0,1,3;0,0,2]');
[VJ,J]=jordan(C)
VJ =
[5, -3, -4]
[3, 0, -3]
[1, 0,  0]
J =
[2, 0, 0]
[0, 1, 1]
[0, 0, 1]
```

【例 3】在自动控制的多变量频域法研究,常要研究传递函数的奇异值分解。本例中的 G 阵就是某系统的开环传递矩阵。

```
G=sym('[1/(s+1),1/(s+4);2/(s+1)/(s+3),-3*s/(s+2)/(s+4)]');
                                        % 传递矩阵 G 的输入指令
SG=simple(singvals(G))                  % 求带复频变量"s"的 G 的奇异值
SN0=simple(subs(SG,'0','s'))            % 求"s"为零时的准确奇异值
SVS=vpa(SN0,20)                         % 求"s"为零时 20 位浮点表示的奇异值
SG =
[1/2*2^(1/2)/(s+4)/(s+3)/(s+2)/(s+1)*(868+393*s^4+924*s^3+1590*s^2
```

```
+1764*s+11*s^6+102*s^5+(5*s^6+20*s^5-45*s^4-226*s^3+66*s^2+836
*s+676)^(1/2)*(17*s^6+184*s^5+831*s^4+2074*s^3+3114*s^2+2692*s+
1060)^(1/2))^(1/2)]
[1/2*2^(1/2)/(s+4)/(s+3)/(s+2)/(s+1)*(868+393*s^4+924*s^3+1590*s^2
+1764*s+11*s^6+102*s^5-(5*s^6+20*s^5-45*s^4-226*s^3+66*s^2+836
*s+676)^(1/2)*(17*s^6+184*s^5+831*s^4+2074*s^3+3114*s^2+2692*s+
1060)^(1/2))^(1/2)]
SN0 =
[1/48*2^(1/2)*(868+676^(1/2)*1060^(1/2))^(1/2)]
[1/48*2^(1/2)*(868-676^(1/2)*1060^(1/2))^(1/2)]
SVS =
[1.2199508581708210995]
[.13661752483748776611]
```

说明：一般地讲，对任何矩阵A都可求得一个奇异分解三对组(U，S，V)，使$A=USV^T$。该分解所得的S称为奇异值阵，而U,V分别称为左右奇异向量阵。在此，要特别说明：由于奇异向量并不随矩阵元素连续变化，因此对于带自由变量的矩阵无法计算奇异向量，而只能给出奇异值。

4.6 符号微积分

在这一节中着重介绍求符号和、符号导数、符号积分、泰勒级数，以及符号雅可比矩阵的求取。具体指令如下：

symsum(S, v)	关于指定变量v对通项S求“不定和”。
symsum(S, v, a, b)	指定变量v在[a, b]之间取值时，对通项S求和。
symdiff(S, v)	计算S对指定变量v的一阶导数。
symdiff(S, v, n)	计算S对指定变量v的n-阶导数。
int(S, v)	计算S对指定变量v的不定积分。
int(S, v, a, b)	计算S对指定变量v在(a, b)区间内的定积分
taylor(F, v)	求F对变量v的麦克劳林展开，6阶小量以余项形式给出。
taylor(F, v, n)	求F对变量v的麦克劳林展开，n阶小量以余项形式给出。
jacobian(F, v)	F是向量函数，是指定变量构成的向量。

说明：上述指令中的“v”可以缺省。缺省时，运算将对默认变量进行。

【例1】求无限项级数的和。

```
symsum('log(a)^k/k!  *x^k','k','0','inf')
ans =
exp(log(a)*x)
```

【例 2】求$\frac{\partial}{\partial x \partial y}\begin{bmatrix} x\sin(y) & x^n + y \\ \frac{1}{xy} & e^{ixy} \end{bmatrix}$

```
S=sym('[x*sin(y),x^n+y;1/x/y,exp(i*x*y)]');
dSdxdy=symdiff(symdiff(S,'x'),'y')
dSdxdy =
[    cos(y),                                    0]
[ 1/x^2/y^2, i*exp(i*x*y)-y*x*exp(i*x*y)]
```

说明：有两个指令可实现符号求导：diff 和 symdiff 。前者既可以求数值差分，又可以求符号导数。而求符号导数时，前者是调用后者进行的。它们的使用格式相同。

【例 3】求$\iint xe^{-xy}dxdy$。

```
(FI=int(int('x*exp(-x*y)','x'),'y')
FI =
1/y*exp(-y*x)
```

【例 4】求$\sin(x)e^{-x}$的麦克劳林级数

```
FT=taylor('sin(x)*exp(-x)',8)
FT =
1*x-1*x^2+1/3*x^3-1/30*x^5+1/90*x^6-1/630*x^7+O(x^8)
```

【例 5】求$[xy(z) \quad y\log z \quad \sin(x)e^{y/z}]^T$对$[x \quad y \quad z]^T$的雅克比矩阵。

```
F=sym('[x*y*z;y*log(z);sin(x)*exp(y/z)]');
v=sym('[x;y;z]');
FJ=jacobian(F,v)
FJ =
[                y*z,                x*z,                      x*y]
[                  0,             log(z),                      y/z]
[ cos(x)*exp(y/z),  sin(x)/z*exp(y/z),  -sin(x)*y/z^2*exp(y/z)]
```

4.7 符号代数方程的求解

在这一节里要介绍两个内容：线性方程组(Linear equations)的符号解(Symbolic solutions)和一般代数方程组(Algebraic equations)的符号解。

4.7.1 线性方程组的符号解

本小节只讨论 A 阵至少行满秩的线性方程组 A * X=B 的解。下面是指令的使用格式:

X=linsolve(A,B) 只给出特解(Particular solution)。

[X,Z]=linsolve(A,B) 将给出由 X 及 Z 构成的通解(General solution)。

说明:

(1)当 A 阵列数大于行数时,将给出解不唯一的警告提示。

(2)X 是方程组的一个特解,Z 是 A 阵零空间的基。通解形式是(X+p * Z),式中 p 是可任意取值的自由参数。

【例 1】求恰定线性方程组的解。

```
A=sym('[1,1/2,1/3;3,1,1;1,2,1]');
B=sym('[1,2;1/3,1;1,1/7]');
X=linsolve(A,B)
X =
[   7/3,          38/7]
[  16/3,            10]
[   -12,        -177/7]
```

【例 2】求欠定方程组的通解。

```
A=sym('[1,1/2,1/3;3,1,1]');
B=sym('[1;1]');
[x,z]=linsolve(A,B)
Warning: Matrix is rank deficient; solution is not unique.
(这是计算中所给的警告信息:解不唯一。)
x =
[  0]
[  4]
[-3]
z =
[  1]
[  0]
[-3]
```

4.7.2 一般代数方程组的解

本小节所讲的 solve 指令能解的一般代数方程包括线性(Linear)方程、非线性(Nonlinear)方程和超越方程(Transcedental equation)。当方程组不存在符号解时,若又无其他自由参数,

则 solve 将给出数值解。具体使用格式如下：

solve(S)　　对一个方程的默认变量求解。

solve(S, v)　　对一个方程的指定变量 v 求解。

solve(S1, S2, …, SN)　　对 N 个方程的默认变量求解。

solve(S1, S2, …, SN, v1, v2, …, vn)　　对 N 个方程的 v1, v2, …, vn 变量求解。

[x1, x2, …, xn] = solve(S1, S2, …, SN)　　对默认变量求解的结果赋给 x1, x2, xn。

[x1, x2, …, xn] = solve(S1, S2, …, SN, v1, v2, …, vn)　　对 v1, v2, …, vn 求解的结果赋给 x1, x2, xn。

【例 1】矩阵特征多项式的根。

```
A=sym('[1,2,1/3;0,5,0;7,0,a]');
CA=charpoly(A)                          % 求 A 阵的特性多项式 CA
RA=solve(CA)                            % 求特性多项式 CA 的根 RA
CA =
x^3-a*x^2-6*x^2+6*x*a+8/3*x-5*a+35/3
RA =
[                                          5]
[1/2*a+1/2-1/6*3^(1/2)*(3*a^2-6*a+31)^(1/2)]
[1/2*a+1/2+1/6*3^(1/2)*(3*a^2-6*a+31)^(1/2)]
```

【例 2】求三元非线性方程组的解。

```
S1='x^2+sqrt(2)*x+1=0';S2='x+3*z=4';S3='y*z=-1';
[x,y,z]=solve(S1,S2,S3)
xv=vpa(allvalues(x),6)
yv=vpa(allvalues(y),6)
zv=vpa(allvalues(z),6)          % 以上三个指令给出符号解的 6 位浮点近似
x =
RootOf(_Z^2+2^(1/2)*_Z+1)
y =
3/(RootOf(_Z^2+2^(1/2)*_Z+1)-4)
z =
-1/3*RootOf(_Z^2+2^(1/2)*_Z+1)+4/3
xv =
[-.707105+.707105*i]
[-.707105-.707105*i]
yv =
[-.623268-9.36279e-2*i]
[-.623268+9.36279e-2*i]
zv =
```

```
[1.56903-.235702*i]
[1.56903+.235702*i]
```

说明:

(1) 式中 RootOf(…)表示括号内表达式的根,采用这种符号的目的是为了更简洁地解析表示各个解及它们之间的关系。符号解常采用这种表达方式。

(2) allvalues 指令用来具体求出 RootOf(…)的全部根的符号解。

(3) vpa 则给出符号解的浮点近似。

【例 3】求解超越方程组。

```
[x,y]=solve('x^x=1/7','x/y=3')
x =
[      exp(lambertw(-log(7)))]
[1/3*exp(lambertw(-log(7)))]
y =
[    1/3*exp(lambertw(-log(7)))]
[ 1/3*exp(lambertw(0,-log(7)))]
```

说明:在上述符号解中的 lambertw(…)代表 ω 函数 $\omega(x)e^{\omega(x)}=x$ 的解。有关内容请用 mhelp 指令去访问 MAPLE 库的 lambertw 条目。

4.8 符号微分方程的求解

本节介绍常微分方程的计算机求解指令 dsolve 。该指令的使用格式为

[y1,y2,…]=dsolve(a1,a2,…,a12)

说明:

(1) 输入宗量包括三部分内容:微分方程、初始条件、指定独立变量,其中微分方程是必不可少的输入内容。后两个内容视需要而定,可有可无。

(2) 关于指定独立变量的规定:若要指定独立变量,则总是由全部输入宗量 a1, a2, ……中的最后一个宗量定义。若不对独立变量加以专门的定义,则本指令默认小写英文字母 x 或 t 为独立变量。

(3) 微分方程的记述规定:当 y 是"应变量"时,用"Dny"表示"y 的 n 阶导数"。比方

Dy　　表示形如$\frac{dy}{dx}$或$\frac{dy}{dt}$的 y 的一阶导数;

Dny　　表示形如$\frac{d^ny}{dx^n}$或$\frac{d^ny}{dt^2}$的 y 的 n 阶导数。

(4) 关于初始条件的规定:初始条件被写成 y(a)=b , Dy(c)=d 等。a, b, c, d 可以是变量使用符外的其他字符。当初始条件少于微分方程数时,在所得解中将出现任意常数符 C1,C2,……解中任意常数符的数目等于所缺少的初始条件数。

(5) 输出宗量可有可无。当有输出宗量时,MATLAB 工作内存中将在 y1, y2,……定义

的输出宗量中保留计算结果。

(6) 输入宗量 a1, a2,……的个数不超过 12 个。但这并不意味着,该指令能解的联立微分方程数受此 12 限制。因为,一个输入宗量可一次定义多个微分方程或多个初始条件(见例)。

【例 1】求$\frac{dx}{dt}=y, \frac{dy}{dt}=-x$的解。

```
[x,y]=dsolve('Dx=y,Dy=-x')
x =
C1*sin(t)+C2*cos(t)
y =
C1*cos(t)-C2*sin(t)
```

说明:(1)t 是默认的独立变量;(2)C1,C2 是任意常数。

【例 2】求$\frac{df}{dx}=3f+4g, \frac{dg}{dx}=-4f+3g, f(0)=0, g(0)=1$的解。

```
[f,g]=dsolve('Df=3*f+4*g,Dg=-4*f+3*g','f(0)=0,g(0)=1')

f =
exp(3*x)*sin(4*x)
g =
exp(3*x)*cos(4*x)
```

【例 3】求解三阶微分方程$\frac{d^3y}{dt^3}=-y$,初始条件为$y(0)=1, \frac{dy(0)}{dt}=0, \frac{d^2y(0)}{dt^2}=0$。

```
y=dsolve('D3y=-y','y(0)=1,Dy(0)=0,D2y(0)=0','t')
y =
1/3*exp(-t)+2/3*exp(1/2*t)*cos(1/2*3^(1/2)*t)
```

4.9 符号函数的二维图形

通过对符号函数的数值计算,利用下一章将介绍的 MATLAB 图形指令,很容易对符号函数进行可视化表现。因此,本节不是对可视化指令作全面介绍,而是专门介绍一个使用十分简便的一元符号函数作图指令 ezplot 。其基本使用格式如下:

ezplot(F,[xmin,xmax],fig)

说明:

(1) 输入宗量 F 是待绘的符号函数。

(2) [xmin,xmax]是界定绘图的自变量范围,可以缺省,缺省值为$[-2\pi,2\pi]$。

(3) 输入宗量 fig 是指定图形窗的，缺省时默认当前图形窗。

【例 1】绘制上节微分方程例 3 所得解函数 y(t)的图形(图 4 10-1)。

```
ezplot(y)                % 绘制符号函数 y(t) 在[-2π,2π]中图形的指令
```

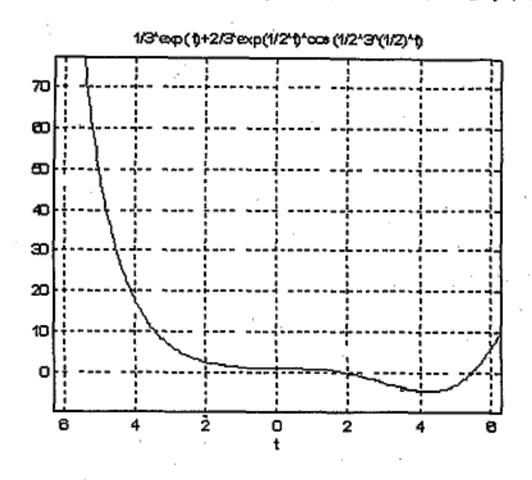


图 4 10-1　符号函数图形

说明：图中的图名(Title)和横坐标轴名都是自动生成的。

4.10　符号计算能力的进一步开拓

相对 MAPLE 软件的 2000 余条符号计算指令而言，前面九节所介绍的内容仅利用了 MAPLE 最常用计算指令中的一部分。为了在 MATLAB 工作环境中，进一步利用 MAPLE 的其他符号计算能力，本节将介绍把 MATLAB 与 MAPLE 联接起来的两个重要指令 maple 和 mpa 。前者用于调动 MAPLE 的符号计算“引擎”和它庞大的函数库，并把最终的计算结果送回 MATLAB 工作间；后者用于向 MAPLE 工作内存输送 MATLAB 中已有的变量内容。

与前面一些符号计算指令相比，maple 的调用要求 MATLAB 用户对 MAPLE 软件有更多的理解和认识。

4.10.1　直接调用 MAPLE 的符号计算能力

在 MATLAB 环境下，为了实现对 MAPLE 绝大多数符号计算指令的调用，该符号数学工具包提供了一个通用指令 maple 。该指令的主要使用格式如下：

maple(MapleStatement)　　调用 MAPLE 中的完整指令 MapleStatement

maple(a0, a1, a2, …, a10)　　调用 MAPLE 中名为 a0 的指令

说明：a0 是字符串；a1, a2, ……是 a0 指令使用格式中依次相应的输入宗量。

【例 1】求递推方程 $f(n)=-3f(n-1)-2f(n-2)$ 的通解。

调用格式一：

```
maple('rsolve(f(n)=-3*f(n-1)-2*f(n-2),f(k))')
ans =
(2*f(0)+f(1))*(-1)^k+(-f(0)-f(1))*(-2)^k
```

调用格式二：

```
maple('rsolve','f(n)=-3*f(n-1)-2*f(n-2)','f(k)')
ans =
(2*f(0)+f(1))*(-1)^k+(-f(0)-f(1))*(-2)^k
```

说明：

(1) 在现有的符号数学工具包中，MATLAB 虽没有提供解“递推方程”的专用函数，但可以通过 maple 这个通用符号计算接口，直接调用 MAPLE 的 rsolve 来解“递推方程”，从而使 MATLAB 的符号计算能力得到进一步的开拓。

(2) 无论是 maple 的哪种调用格式，都需要对 MAPLE 的相关指令有直接的认识和理解。

【例 2】求 f=xyz 的 Hessian 矩阵。

```
FH=maple('hessian(x*y*z,[x,y,z])')
FH =
[0, z, y]
[z, 0, x]
[y, x, 0]
```

说明：

(1) 由于 maple m 文件编写上的原因，对于像本例这种在一个参数位置上含有方括号的多变数的情况，maple 的调用格式二将无法运用。有兴趣的读者不妨试试。

(2) 请记住：maple 的调用格式一对 MAPLE 软件指令的适应性要比调用格式二强得多。

【例 3】求 $\sin(x^2+y^2)$ 在 x=0, y=0 处展开的截断 8 阶小量的泰勒近似式。

过程一：

```
maple('mtaylor(sin(x^2+y^2),[x=0,y=0],8)')
        % 在此，mtaylor(sin(x^2+y^2),[x=0,y=0],8)是完整的 MAPLE 指令。
ans =
mtaylor(sin(x^2+y^2),[x = 0, y = 0],8)
```

所得显示表明，展开并没有进行。

过程二：

```
maple('readlib(mtaylor);');
maple('mtaylor(sin(x^2+y^2),[x=0,y=0],8)')
ans =
x^2+y^2-1/6*x^6-1/2*y^2*x^4-1/2*y^4*x^2-1/6*y^6
```

所得的展开结果正确。

说明：

(1) MAPLE 软件所有的函数，在初始化时并没有全部装入内存。因此，有些“读库函数

(Readlib－defined functions)"在第一次调用前，需要先运用 MAPLE 的 readlib 指令把该函数由库读入内存。readlib 在 MATLAB 中的使用格式是

maple('readlib(FunName);')　　FunName 为将按装的函数名(如例 3 的 mtaylor)。

(2) 读者会注意到，本例过程一的"运算显示结果"恰是输入的 MAPLE 指令本身。这一现象提示用户，所用的函数也许还没有读入。

(3) mtaylor 指令不仅能求在 $x=0, y=0$ 处泰勒展开，而且能求在 $x=x_0, y=y_0$ 处的一般展开式。

4.10.2　给 MAPLE 工作空间中的变量定义

MATLAB 和 MAPLE 有各自的工作空间，在这两个不同工作空间中的变量，不管它们的名称是否相同，它们是独立的，不相关的。在前面讲述的符号计算中，用户也许没有感觉到这两个独立工作空间的存在，那正是 MATLAB 符号数学工具包的功效。

本小节要介绍一个向 MAPLE 工作空间输送变量的指令 mpa 。该指令与 maple 配合使用，将使用户能在 MATLAB 的工作环境下更充分地发挥 MAPLE 的符号计算功能。

mpa 的调用格式如下：

mpa('v',S)　　在该调用格式中，v 必须在引号中。

mpa v S　　该调用格式中的空格是必须的。

在此：v 是 MAPLE 工作空间中的被定义变量名；S 本身是字符串或是 MATLAB 工作空间中已存在的字符串变量名。

【例 1】求 $f(k)=e^{-2k}$ 的 Z 变换 $F(z)=\sum_{k=0}^{\infty} f(k)z^{-k}$。

```
f='exp(-2*k)';                    % 在 MATLAB 工作空间中定义字符串变量 f
maple('readlib(ztrans);');        % 把 ztrans 函数从 MAPLE 的库中读人它的内存
mpa('v',f);                       % 用 MATLAB 变量 f 定义 MAPLE 中的变量 v
FZ=maple('ztrans(v,k,z);')        % 把 Z 变换结果赋给 MATLAB 空间的 FZ 变量
FZ =
z/(z-exp(-2))
```

说明：

(1) 以上运算中所用到的变量 f, FZ 都是 MATLAB 中的变量。而 v 是 MAPLE 中的变量。有兴趣的读者可以在 MATLAB 环境中，用 who 指令查看 MATLAB 工作空间中有没有 v。

(2) 上述 ztrans 指令中被变换的函数变量名 v 必须靠 mpa 去定义。当然，对于像本例这样比较简单的被变换函数可不必这样做，而采用如下的直接输入法去求 Z 变换：

FZ=maple('ztrans(exp(-2*k),k,z);')

(3) 在 maple 的调用格式二中，MATLAB 中的变量可直接送入 MAPLE 空间。因此，上述解算 Z 变换的过程也可由以下指令实现：

```
f='exp(-2*k)';
maple('readlib(ztrans);');
```

```
FZ=maple('ztrans',f,'k','z')
```

【例2】帮助理解不同工作空间中的变量关系。

```
mpa('a','1 05');mpa('b','sqrt(1/2)');
aa=maple('a'),bb=maple('print(b)')
A=str2num(aa);B=str2num(bb);
A+B

aa =
1.05
bb =
1/2 * 2^(1/2)
ans =
    1.7571
```

说明：

(1) 在上述过程中，a、b是MAPLE中的变量名；aa、bb是MATLAB中的字符串变量名；A,B是MATLAB中的数值变量名。

(2) maple还有把MAPLE中变量内容传送到MATLAB空间的功能。

(3) maple('print(b)')中的print把变量b的内容以较“好看”的形式显示在屏幕上。

4.11 图示化函数计算器

本节要介绍MATLAB符号数学工具包所提供的第三种符号计算方式——图示化函数计算器。它的外形如图4 11-1所示。该图示化函数计算器是由funtool m文件生成的，即在MATLAB环境下，输入以下指令便可·

funtool

该图示化函数计算器由两个函数曲线视窗(图4 11-1中的Figure No 1和Figure No 2)和一个函数运算控制器(图4.11-1中的Figure No 3)组成。

4.11.1 函数曲线视窗的激活

任何时候，这两个函数曲线视窗中只有一个是活的，或是Figure No.1，或是Figure No.2。在图4 11-1中，Figure No.2正处于活动状态。每个函数曲线视窗的激活由在该视窗左下角的【f1】(或【f2】)控制。假如，在图4 11-1所示情况下，用鼠标点击按键【f1】，那么函数曲线视窗Figure No 1将成为活动窗，与此同时关闭函数曲线视窗Figure No 2。

函数运算控制器上的任何操作都只对活动的函数曲线视窗起作用。换句话说，随运算控制器上的不同操作，活动的函数曲线视窗中的图示曲线将作相应的变化。

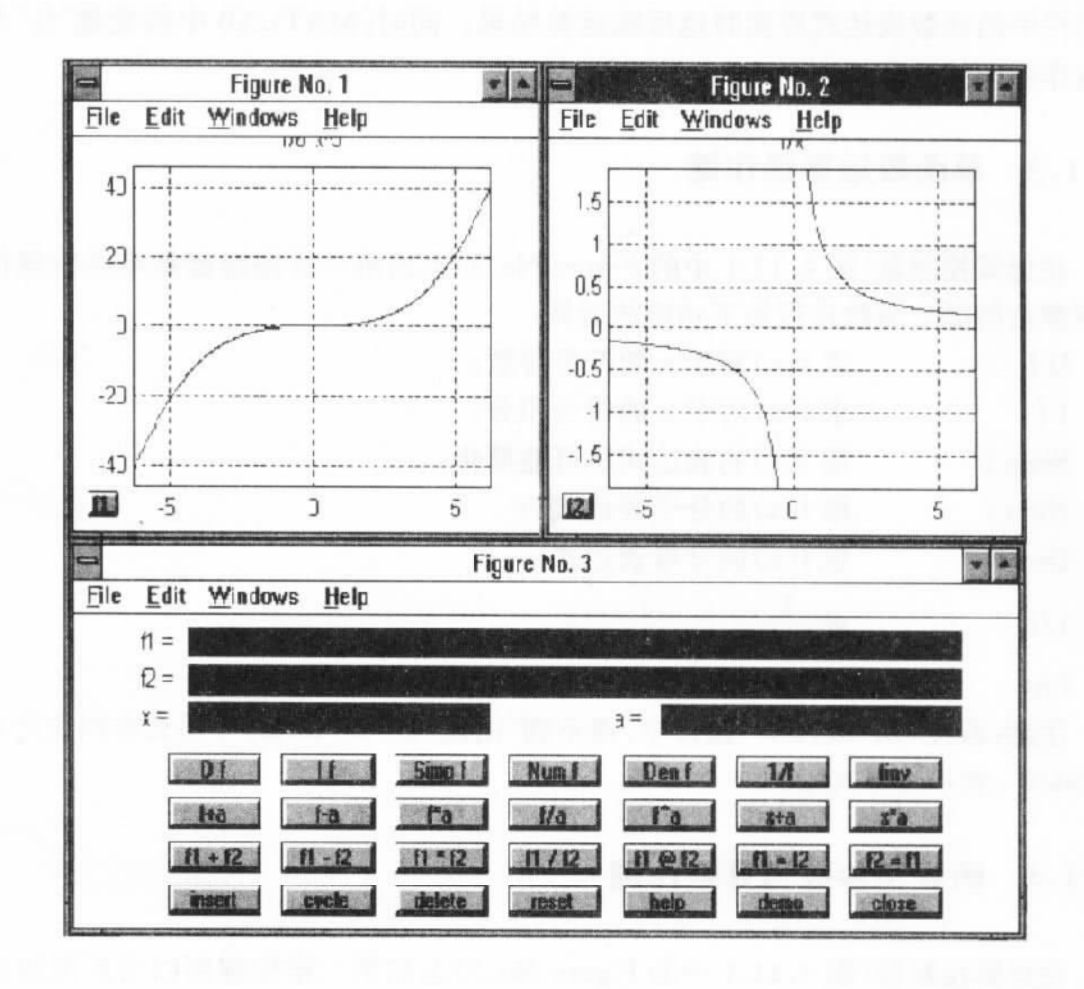


图 4.11-1 图示化函数计算器

4.11.2 运算控制器上被控栏的操作

被控栏是指在函数运算控制器(图 4.11-1 中的 Figure No.3)上半部分的那四个栏目："f1"，"f2"，"x"，"a"。"f1"，"f2"栏分别显示着相应视窗 Figure No.1 和 Figure No.2 中所视曲线的函数表达式；"x"栏显示着函数曲线视窗中作为横坐标变量的取值范围；"a"栏显示着可能参与函数运算的自由参数值。

被控栏内容有以下三种来源和操作方式：

(1) 在图示化函数计算器打开之前，若 MATLAB 工作空间中已有同名字符串变量"f1"，"f2"，"x"，"a"存在，那么这些变量的内容会被即将打开的图示化函数计算器所调用。反之，若 MATLAB 中没有已定义的同名变量，则新打开的图示化函数计算器将默认 f1 = 'x'， f2 = '1'， x = '[- pi, pi]'， a = '1/2'。

(2) 在已打开的图示化函数计算器上，除"x"栏以外，被控栏"f1"，"f2"，"a"中的内容可以用下述方法随时修改。移动鼠标使光标移到要修改内容之处，通过键盘操作就可写入用户所希望的内容。在以上操作结束之后，依次激活相应的函数曲线视窗，便可看到修改后所得的函数曲线。

(3) 当用运算控制器进行运算操作时，被激活的曲线视窗中的函数曲线和该视窗对应的

被控栏中的函数表达式将实时地反映运算结果。同时，MATLAB 中的变量“f1”和“f2”中的内容也作相应的改变。

4.11.3　单函数运算操作键

在运算控制器(图 4.11-1 中的 Figure No 3)上的第一排按键都是单函数操作键。它们只对所激活的那个函数进行如下功能的运算：

D f　　求 f(x)对于 x 的符号导数。

I f　　求 f(x)对于 x 的符号积分。

Simp f　　使 f(x)的表达式尽可能简化。

Num f　　取 f(x)的分子表达式。

Den f　　取 f(x)的分母表达式。

1/f　　求$\frac{1}{f(x)}$。

finv　　求 f(x)的反函数 g，使 g(f(x))=x。

注意：若在“I f”或“finv”操作中，得不到“闭式(Closed form)”，那么在相应的函数栏里将给出“NaN”，表示运算失败。

4.11.4　函数和参数运算操作键

在运算控制器(图 4.11-1 中的 Figure No 3)上的第二排按键用以实现激活函数和自由参数 a 的运算操作。具体如下：

f+a　　计算 f(x)+a。

f-a　　计算 f(x)-a。

f*a　　计算 af(x)。

f/a　　计算 f(x)/a。

f^a　　计算 $f^a(x)$。

x+a　　计算 f(x+a)。

x*a　　计算 f(ax)。

4.11.5　两个函数间的运算操作键

运算控制器上的第三排按键用以实现两个函数间的运算。运算结果一方面图示在激活函数窗里，另方面把所得的结果表达式显示在相应的被控函数栏里。运算操作键的具体功能如下：

f1+f2　　求 $f_1(x)+f_2(x)$

f1-f2　　求 $f_1(x)-f_2(x)$

f1*f2　　求 $f_1(x)f_2(x)$

f1/f2　　求$\frac{f_1(x)}{f_2(x)}$

f1@f2	求复合函数 $f_1(f_2(x))$
f1=f2	用 $f_2(x)$ 取代 $f_1(x)$
f2=f1	用 $f_1(x)$ 取代 $f_2(x)$

4.11.6 辅助操作键

在运算控制器第四排上七个按键的功能如下：

insert	把当前 Figure No 1 视窗里的函数插入到内含的典型函数演示表中。
cycle	在 Figure No 1 视窗里依次显示内含的典型函数演示表中的函数曲线。
delete	从内含的典型函数演示表中删除当前 Figure No 1 视窗中的函数。
reset	把整个函数计算器重置成初始调用状态。
help	在 MATLAB 窗里给出函数计算器的在线帮助。
demo	将自动演示函数计算器的运算功能。
close	关闭函数计算器。

4.12 符号计算指令的在线求助

关于符号计算指令的在线求助要分两个层次来介绍。第一层次是关于 MATLAB 符号数学工具包中所有 M 文件的在线求助；第二层次是关于 MAPLE 库函数指令的在线求助。

4.12.1 MATLAB 符号数学工具包中 M 文件的在线求助

与 MATLAB 其他工具包在线求助方法相同，获取 MATLAB 符号数学工具包中 M 文件的在线帮助信息有以下几种手段：

(1) 利用"dir c：\ matlab \ toolbox \ symbolic"指令列出符号数学工具包中的所有文件。从中可得知：该工具包中 M 文件的概貌和各 M 文件的确切名称。

(2) 利用"lookfor symbolic"指令可列出进行符号计算的各 M 文件的名称和简短说明。

(3) 利用"type c：\ matlab \ toolbox \ symbolic \ contents m"获得关于各符号计算 M 文件功能的分类信息。

(4) 利用"help Mname"（注：这 Mname 是指具体的符号计算 M 函数名）获得关于具体函数使用方法的详细说明。

4.12.2 MAPLE 库函数在线帮助的检索树

获得 MAPLE 库函数指令在线帮助的主要手段是"mhelp"，至于前面所介绍的获取符号计算 M 文件信息的"help"、"lookfor"在此几乎完全无用。

下面将沿着检索树自上而下地介绍 MAPLE 的在线求助方式。

(1) mhelp index

该指令将给出 MAPLE 指令的如下归档类目：

library	标准库函数(Standard library functions)指令集。
packages	库函数包(Library packages of functions)集。
libmisc	辅助库函数(Miscellaneous lib functions)指令集。
statements	语句描述(Descriptions for Maple statements)指令集。
expressions	表达式描述(Descriptions for Maple expressions)指令集。
datatypes	数据类型(Descriptions for Maple datatypes)指令集。
tables	列表和数组描述(Descriptions for tables and arrays)指令集。
procedures	程序(Maple procedures)操作指令集。
misc	其他辅助(Miscellaneous facilities)指令。

(2) mhelp index[CategoryName]

该指令中的 CategoryName 是指各类目名，如 library 等。该指令的执行将给出更详细的次下层信息。如 mhelp index[library]，给出次下层树枝名"internal"、"external"、"inert"，又给出了 250 个函数名清单。再比如 mhelp index[package]将给出"geom3d"、"numtheory"等 25 个子包(Subpackage)名称。

(3) mhelp SubName

在此的 SubName 是指次下层树枝名或子包名。如 mhelp internal，将给出内部库函数名清单。又如 mhelp geom3d，将给出有关立体解析几何函数指令的清单。

(4) mhelp Fname

这里 Fname 是指具体的 MAPLE 函数名称。该 mhelp 将给出关于该指令的详细说明(包括定义、使用格式、举例说明及相关指令)。MAPLE 函数的确切名称可以用前面讲的方法在线查阅。

4.12.3 MATLAB 提供的 MAPLE 特殊函数名清单文件

MATLAB 为用户查询方便，编制了一个纯说明文件"mfunlist"。该文件不作任何实质性的运算，而仅采用 MATLAB 注释语句列出了 50 多种常用的 MAPLE 特殊函数名及简单的说明。在 MATLAB 环境下，help mfunlist 和 type mfunlist 的效果相同。

4.13 补充说明

本节罗列 MATLAB 符号数学工具包中几个尚未涉及的指令。更详细的使用说明请看用户手册或借助 help 获得在线帮助信息。

(1) compose

求复合函数指令。

(2) finverse

求反函数指令。

(3) latex

向 LaTex 字处理软件提供数学表达式的"精美书写"指令。

(4) pretty

在英文 MATLAB 环境中能给出较漂亮的数学书写形式。但在中文 MATLAB 环境，由于某些制表符的 ASCII 码被汉字所占用，因此该指令有可能给出不伦不类的书写形式。

第五章　计算结果的可视化

人们很难直接从一大堆原始的离散数据中感受到它们的含义，数据图形恰能使人们用视觉器官直接感受到数据的许多内在本质。因此，数据可视化是人们研究科学、认识世界所不可缺少的手段。

作为一个优秀的科技应用软件，MATLAB 不仅数值计算方面无与伦比，而且在数据可视化方面也有上佳表现。

MATLAB 可以给计算数据以二维、三维乃至四维的图形表现。通过对图形线型、立面、色彩、渲染、光线、视角等品性的处理，可把计算数据的特征表现得淋漓尽致。

MATLAB 图形系统的这种能力是建立在一组“图形对象(Graphics Objects)”基础之上的。它的核心是图形的句柄(Graphics Handle)操作。MATLAB 有两个层次的绘图指令：一组是直接对句柄进行操作的底层(Low－level)绘图指令；另一组是在底层指令基础上建立起来的高层(High－level)绘图指令。

高层绘图指令简单明了。它不仅容易为用户所掌握，而且也是今后最常用的。本章将用较大的篇幅来阐述高层绘图指令的作用、调用格式和规律。本章内容按“前易后难”的原则安排。

底层绘图指令控制和表现数据图形的能力更强。由于它们比较灵活多变，因此这部分内容被安排在本章的最后一节。有 MATLAB 使用经验的用户，通过该节的内容和例题，不难掌握底层指令的运用方法。

为了使初学者能尽快地掌握和了解 MATLAB 的图形功能，本章第一节的内容是一个很好的入门引导。同时，该节的内容和说明也是阅读后几节的基础。因此，即便是对已经会用 MATLAB 图形指令的读者来说，本节也将提供有用的信息。

5.1　入　门

本节介绍两个最常用的绘图指令：plot 和 mesh 。指令 plot 绘制二维曲线，mesh 绘制三维网面。从 3.0 版起，MATLAB 就提供了这两个指令，只是那时的 mesh 指令不表现坐标轴。

5.1.1　二维图形

本书前面各章已经多次用过 plot 指令，读者对此不会觉得陌生。plot 是最基本的二维图形指令，它以 MATLAB 的内部函数(Build－in Function)形式出现。MATLAB 其他二维图形指令中的绝大多数是以 plot 为基础构造的。因此，本节将比较全面地介绍这个指令的用法。

基本操作

下面介绍 plot 指令绘制二维曲线时的基本调用格式。

(1) plot(X)

若 X 为向量，则以 X 元素值为纵坐标，以相应元素下标为横坐标值，绘制连线图。若 X 为实数阵，则按列绘制每列元素值相对其下标的连线图，图中曲线数等于 X 阵的列数；若 X 为复数阵，则分别以 X 实部阵和虚部阵的对应列元素为横纵坐标绘制多条连线图。

(2) plot(X, Y)

若 X、Y 是同维向量，则绘制以 X、Y 元素为横、纵坐标的连线图。若 X 是向量，Y 是有一维与 X 等维的矩阵，则绘出多根不同色彩的连线图，连线根数等于 Y 阵的另一个维数，X 被作为这些曲线的共同横坐标。若 X 是矩阵，Y 是向量，情况与上相同，只是曲线都以 Y 为共同纵坐标。若 X、Y 是同维矩阵，则以 X、Y 对应列元素为横纵坐标分别绘制曲线，曲线条数等于矩阵的行数。

(3) plot(X1, Y1, X2, Y2,)

在此格式中，每个二元对 X - Y 的作用与 plot(X, Y)相同，每个二元对 X - Y 的结构也必须符合 plot(X, Y)的要求。但不同二元对之间没有约束关系。

在以上三种格式里，输入宗量 X、Y 都可以是表达式，只要这表达式的运算结果符合上述格式要求。

【例 1】单向量输入格式，并画连线(图 5 1-1)。

```
X=[1,3,6,3,5,9,0,2];
plot(X)
```

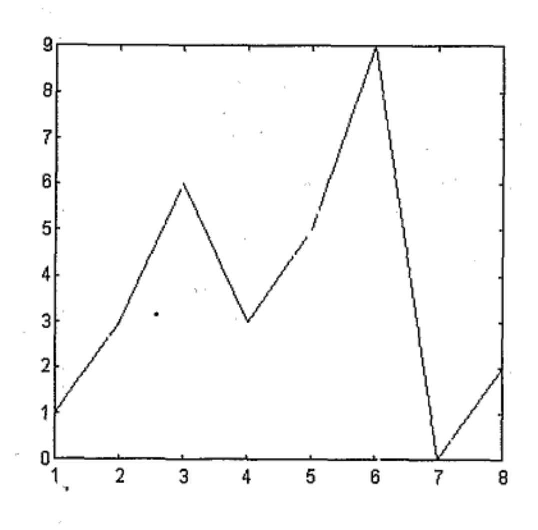


图 5 1-1 单向量输入格式所画连线

【例 2】双向量输入格式，并画曲线(图 5 1-2)。

```
X=0:pi/100:2*pi;
Y=sin(X);
plot(X,Y);
说明.
```

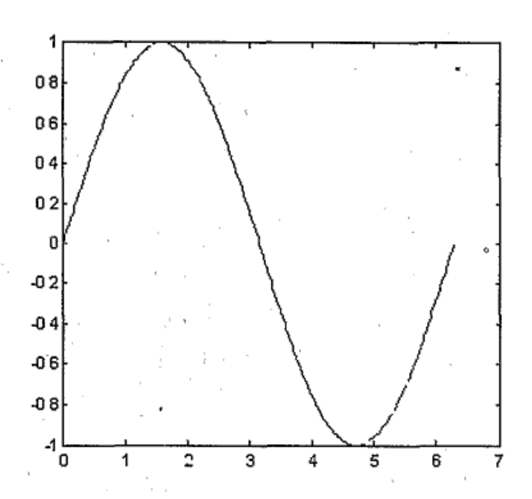


图 5 1-2 双向量输入格式所画曲线

(1) 本例第一条赋值语句使 X 成为行向量。这是最常见的绘图数据生成方式。

(2) 若用 plot(X,sin(X)) 代替该例二、三两句指令,效果完全一样。

【例 3】在输入宗量一个是向量一个是矩阵情况下,以相同横坐标画多根曲线(图 5 1-3)。

```
X=0:pi/100:2*pi;
X=X';
Y=[sin(X),cos(X),cos(X+0.5)];
plot(X,Y);
```

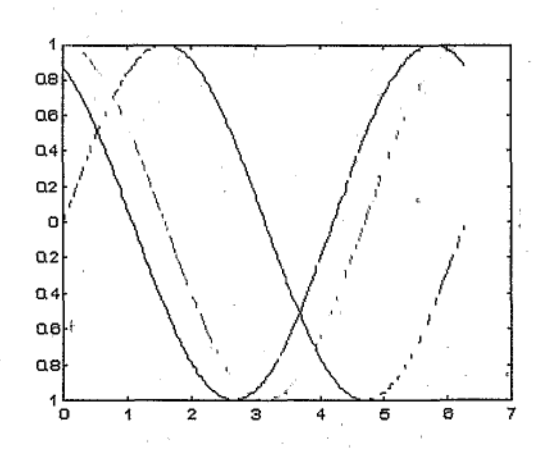


图 5 1-3 以相同横坐标画多条曲线

【例 4】据 Euler 公式:$e^{it}=\cos(t)+i\sin(t)$,利用单输入复数向量画单位圆(图 5 1-4)。

```
t=0:pi/100:2*pi;Y=exp(i*t);
plot(Y);axis('square')
```

说明:最后一条指令 aixs('square') 的作用是,保证所画图形的纵横坐标刻度比例相同。

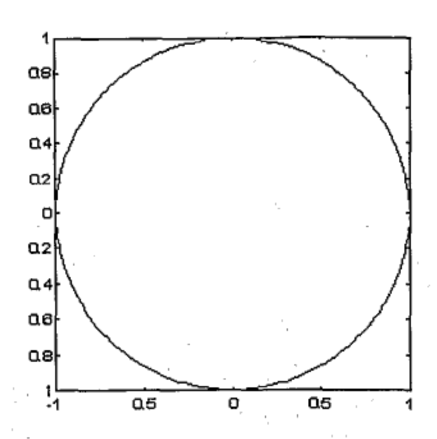


图 5 1-4 单输入复数向量画单位圆

开关控制

二维绘图指令 plot 还提供一组控制曲线线型、标记类型和颜色的开关。该开关总跟在一元或二元对的后面，具体如下：

plot(X, S)

plot(X, Y, S)

plot(X1, Y1, S1, X2, Y2, S2,)

其中，S 是用单引号标记的字符串。该字符串由 1～3 个表 5 1-1 中的字符组成。

表 5.1-1 曲线线型、标记类型、颜色的控制字符

色彩字符	指定色彩	绘图字符	指定绘图形式
y	黄		小黑点(标数据用)
m	洋红	o	小圈号(标数据用)
c	青	x	叉号(标数据用)
r	红	+	十字号(标数据用)
g	绿	*	星号(标数据用)
b	蓝	—	实连线
w	白		虚点连线
k	黑	–	点划连线
		– –	双划连线

注意"绘图字符"上面五栏中的字符，是用来标记一个个孤立数据点的。

【例 5】绘图线型、色彩设定、坐标网线和图形加注。

```
t=0:pi/12:2*pi;
y1=sin(t);y2=cos(t);
plot(t,y1,'r-',t,y2,'b--');          % 线型、色彩设定
grid                                 % 坐标网线
X=[1 7*pi;1.6*pi];
```

```
Y=[-0.3; 0 8];
S=['正弦曲线 sin(x)';'余弦曲线 cos(x)'];
text(X,Y,S);
title('正弦和余弦曲线');
```

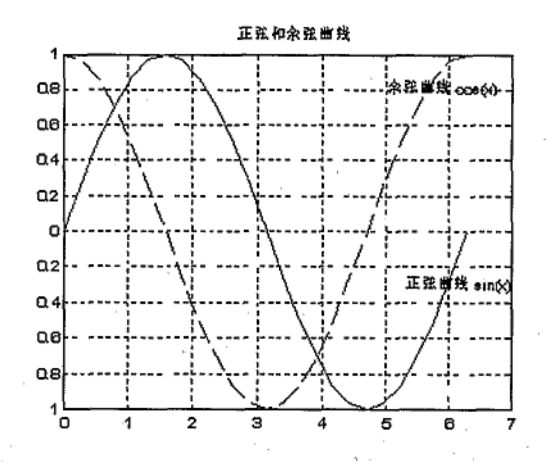


图 5 1-5　绘图线型、色彩设定和图形加注

5.1.2　三维网线图初步

作为入门，本节只介绍绘制三维网线图指令 mesh 的简单用法。

三维网线图的形成原理是：对 x-y 平面某一指定矩形范围采用与坐标轴平行的直线将其分格；计算矩形网格点上的函数值 $z=f(x,y)$，得到 $x-y-z$ 三维空间内的数据点；将这些数据点分别用处于 $x-z$ 或其平行面内的曲线和处于 $y-z$ 或其平行面内的曲线连接，即得 mesh 图。

两个最基本的 mesh 调用格式为：

(1) mesh(Z)

以 Z 矩阵元素值及其下标为数据点，绘制网线图。

(2) mesh(X, Y, Z)

若 X、Y 是向量，那么必须有：X 的长度＝矩阵 Z 的列维；Y 的长度＝矩阵 Z 的行维。网格数据点的 $x-y$ 坐标由 X、Y 向量元素组合构成，纵坐标则取自矩阵 Z。

若 X、Y、Z 是同维矩阵，则数据点的坐标分别取自这三个阵。

【例 1】单输入宗量时，绘制 mesh 图。

```
z =[
     1    1    1    1    1    1    1
     1    2    2    2    2    2    1
     1    2    2    3    2    2    1
     1    2    2    2    2    2    1
```

```
    1    1    1    1    1    1    1];
mesh(z);colormap([1 0 0]),
xlabel('x 轴');ylabel('y 轴');
```

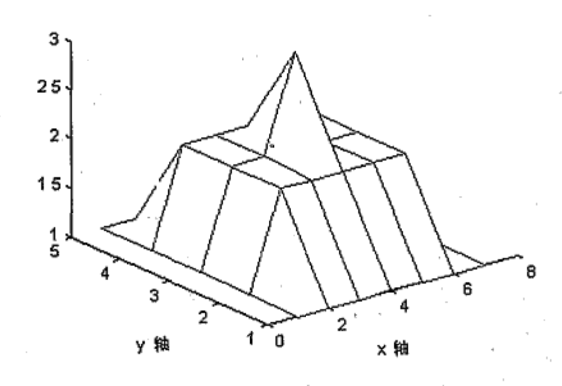


图5 1-6　简单网线图

说明：

(1) 由例可见，网格数据点 x－y－z 的 x 坐标取自矩阵 Z 的列下标，y 坐标取自矩阵 Z 的行下标。

(2) 本例指令中的 colormap([1 0 0])是用来控制网线色彩的，关于它稍后再述。

【例 2】绘制 $z=x^2+y^2$的三维网线图形。

```
clf
x=-4:4;y=x;
[X,Y]=meshgrid(x,y);          %生成网格点的 x－y 坐标
Z=X.^2+Y.^2;
mesh(X,Y,Z);colormap([1 0 0]);hold on
Z0=zeros(size(X));
plot3(X,Y,Z0,'bo')             %绘制网格点在 x－y 平面上的投影
```

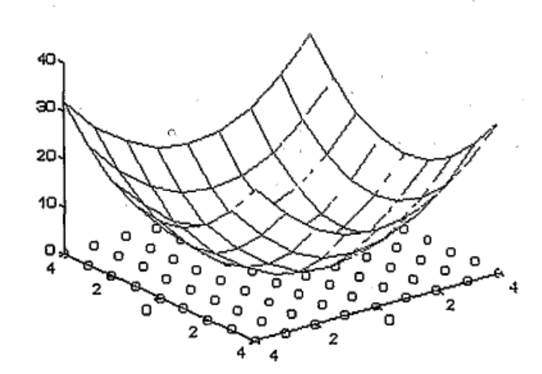


图5 1-7　三维网线及网格点的投影

5.2 曲线图形

除了 plot 指令外，MATLAB 还提供了许多其他二维和三维的曲线图形指令，这些指令大大扩充了 MATLAB 的曲线作图能力，可以满足用户的不同需要。

5.2.1 二维特殊图形

MATLAB 所提供的绘制二维图形的指令很多，详见表 5 2-1 。

上节详细介绍了基本绘图指令 plot 。表中的符号函数二维曲线绘制指令 ezpolt 已在第 4 9 节作了详细介绍。统计频数图形指令 hist 和 rose 的用法请见第 3 10 3 节。下面以举例方式再介绍几个常用的指令。至于它们调用格式的详细介绍请参考用户手册或用 help 功能在线查询。

表 5.2-1 绘制二维图形的指令

bar	直方图	loglog	双对数坐标曲线
compass	原点出发的复数向量图	pcolor	伪彩图
contour	在 x－y 平面上绘制等位线图	polar	极坐标曲线
errorbar	误差棒棒图	plot	直角坐标二维曲线
ezpolt	符号函数二维曲线	quiver	矢量场图
feather	沿 x－轴分布的复数向量图	rose	统计频数扇块图
fplot	数值函数二维曲线	semilogx	x－轴对数坐标曲线
fill	平面多边形填色	semilogy	y－轴对数坐标曲线
gplot	绘拓扑图	stem	火柴杆图
hist	统计频数直方图	stairs	阶梯图

【例 1】用 bar(x, y) 绘制向量 y 的直方图(图 5 2-1)。

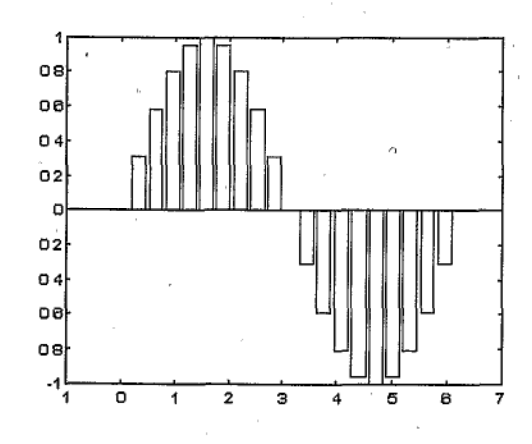


图 5 2-1 直方图

```
x=0:pi/10:2*pi;
y=sin(x),
bar(x,y);
```

说明:

(1) 如果使用指令 bar(y),则取 y 向量元素下标作为 x 坐标值。

(2) 本指令也可像 plot 一样加入指令开关,选择线型和颜色。

【例 2】用 Errorbar(x, y, e) 绘制误差棒棒图。

```
x=0:pi/12:2*pi;
y=sin(x);
e=rand(size(x))/5;
errorbar(x,y,e);
```

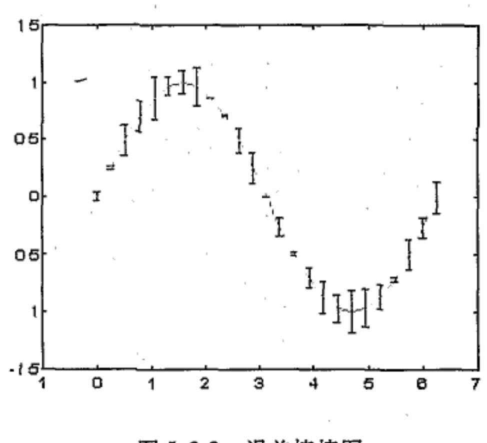


图 5 2-2 误差棒棒图

说明:

(1) x 、y、e 必须是同维。

(2) 该指令常用于数理统计数据的可视化。

【例 3】指令 stem(x, y) 绘制离散数据序列的火柴杆图(图 5 2-3)。

```
x=0:0.2:4*pi;
y=exp(-0.3*x).*sin(x);
stem(x,y)
```

说明:

(1) 若输入指令格式为 stem(y) ,那么取 y 元素下标作为 x 坐标。

(2) 指令格式中,可以带开关,以指定直线的线型和颜色。

【例 4】用极坐标指令 polar(t, y) 绘制螺旋线(图 5 2-4)。

```
t=0:0.01:5*pi;y=t;
polar(t,y)
```

说明：

(1) t 、y 是同维的向量或矩阵，t 的单位是弧度。

(2) 可以用开关控制线型和色彩。

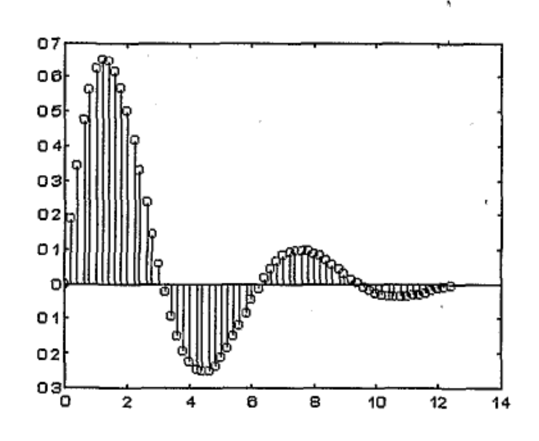


图 5 2-3 火柴杆图

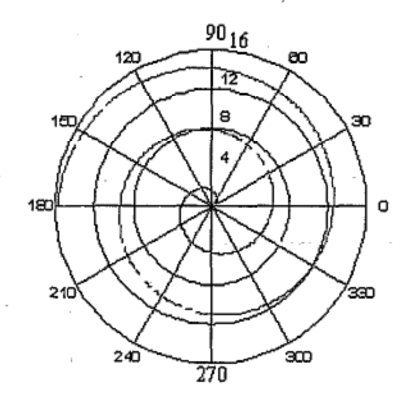


图 5 2-4 螺旋线

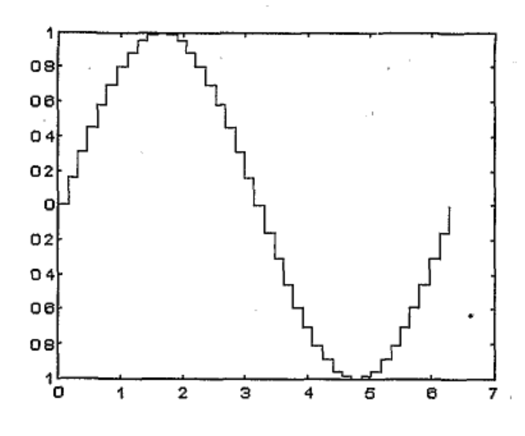


图 5 2-5 阶梯曲线

【例 5】用 stairs 指令绘制阶梯曲线(图 5 2-5)。

```
x=0:pi/20:2*pi;y=sin(x);
stairs(x,y)
```

说明:可以用开关控制线型和色彩。

5.2.2 绘制数值函数二维曲线的专用指令

前面所介绍的绘图指令在绘制一个函数 y=f(x)的图形时,必须先定义自变量 x 的一组取值点,再求出这组取值点上的函数值,然后根据这两组数值确定的数据点绘制出所需的图形。

本节要介绍绘制函数 y=f(x)图形的一个专用指令:fplot 。该指令的特点在于:它的绘图数据点是自适应产生的。在函数曲线平坦处,它所取数据点比较稀疏;在函数变化剧烈处,它将自动取较密的数据点。因而,对于那些导数变化较大的函数,用 fplot 所绘出的曲线比等分取点所画出的曲线更加接近真实。fplot 的用法如下(MATLAB 4 0 版中的 fplot 指令调用格式与此略有不同):

fplot(fname, lims, marker, tol)

[X, Y]=fplot(fname, lims, marker, tol)

其中各项的含义是:

fname	函数名称字符串。它可以是一个由多个分量函数构成的函数行向量。分量函数可以是 MATLAB 已有函数,也可以是用户自定义函数。
lims	定义 x 的取值区间,lims=[xmin, xmax]。
marker	绘图所用的色彩、线型标记,使用方法与 plot 一样,参见第 5 1 1 节。
tol	相对误差,缺省值为 2e-3。在绘图时,它用于控制逐段线性延长与真实函数值之间的误差。
X, Y	输出数据点坐标。X 是横坐标列向量,Y 是按列给出各分量函数的纵坐标。若不指定 X 和 Y,则直接画出图形。

【例 1】fplot 与一般绘图指令的绘图效果比较。

(1) 横坐标自适应取点绘图

为比较需要,自适应绘图分两步进行。

```
[X,Y]=fplot('cos(tan(pi*x))',[-0.4,1.4],0.2e-3);
n=length(X)
plot(X,Y)
n =
        1337
```

(2) 横坐标等分取点绘图

```
t=(-0 4.1 8/n:1 4)';
plot(t,cos(tan(pi*t)));
```

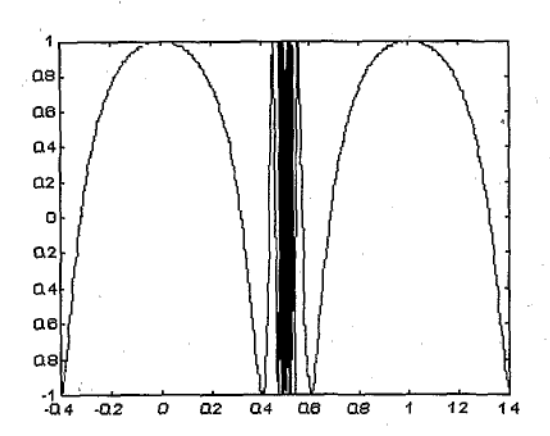


图 5 2-6　横坐标自适应取点绘图

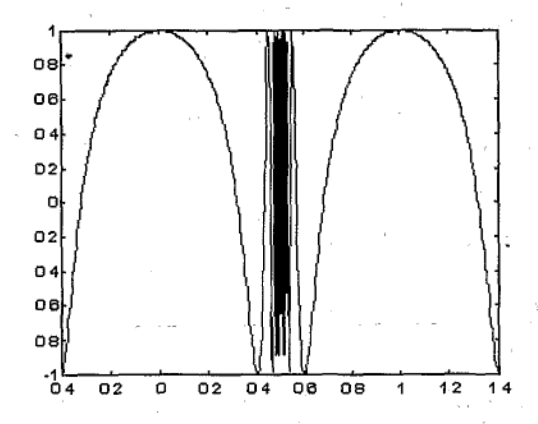


图 5 2-7　横坐标等分取点绘图

说明：

（1）本例中的 n 是自适应绘图时所取的数据点数。

（2）在相同数据点数下，显然自适应取点所绘制的图更真实。

（3）自适应绘图所需的时间较长。

5.2.3　三维曲线

MATLAB 提供了在三维空间中绘制曲线的基本函数 plot3 。像 plot 函数一样，它也是一个 MATLAB 内部函数，因而其他的一些三维曲线函数都要直接或间接地用到该函数。

plot3 的基本调用格式是：

```
plot3(X, Y, Z)
plot3(X, Y, Z, S)
plot3(X1, Y1, Z1, S1, X2, Y2, Z2, S2,      )
```

在此，X、Y、Z 是同维的向量或矩阵。当它们为矩阵时，它们相应的列构成一条三维曲线

的数据点坐标。S 是指定数据点标号类型、曲线线型、色彩的开关。它是表 5 1－1 定义的字符串。

【例 1】指定线型、色彩的三维曲线

```
t=0:0 05:100;
x=t;y=sin(t);z=sin(2*t);
plot3(x,y,z,'b·');
```

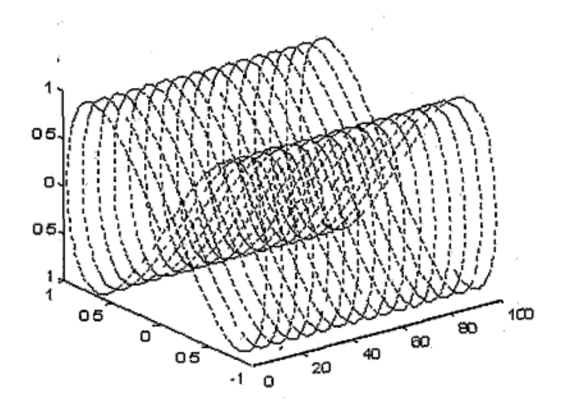


图 5 2-8　指定线型、色彩的三维曲线

5.2.4　多边形的填色

填色是指，将多边形的内部涂满指定的颜色。MATLAB 提供两个填色函数：fill 和 fill3 。前者用于平面多边形的填色，后者用于空间多边形的填充。多边形可以是凸的，也可以是凹的，并允许多边形的边自相交。如果填色时，为每个点指定颜色，MATLAB 将采用双线性插值决定其内部的颜色，关于这点将在第 5 7 4 节解释。

<u>二维多边形</u>

二维填色指令的调用格式如下：

fill(x, y, c)

fill(x1, y1, c1, x2, y2, c2, 　)

说明：

(1) x , y (或 x1 , y1 等)是同维向量。按向量元素下标渐增次序依次用直线段连接 x, y 对应元素定义的数据点。倘若这样连接所得折线不封闭，那么 MATLAB 将自动把该折线的首尾连接起来，构成封闭多边形。

(2) c (或 c1 等)若取表 5 1－1 中色彩字符串，则该多边形用该字符所代表的颜色填充。若取 RGB 三元行向量，则该多边形用该 RGB 三元行向量所调之色填充。若取与 x, y 同维向量，则用 c 向量元素在当前色板上所对应的色彩定义相应数据点(即多边形顶点)的颜色，而该多边形内部颜色据它们插补而成。

(3) 调用格式中的三元对(x , y , c)也可以是矩阵,请看用户手册或在线帮助。

【例 1】单色填充时,填色指令工作原理示意。

```
x=[1,2,3,4,5];y=[3,5,2,1,6];
fill(x,y,'r');hold on
plot(x,y,'ro')
```

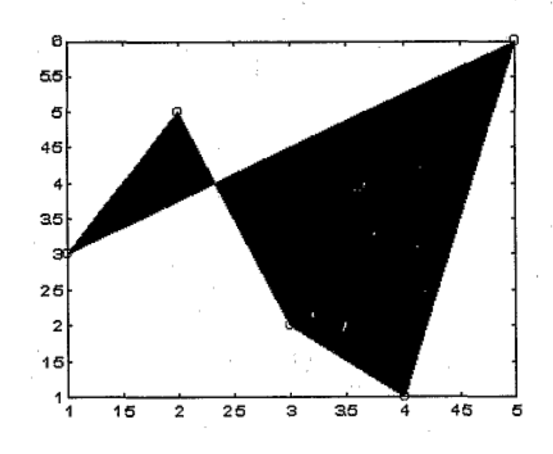


图 5 2-9　单色填充工作原理示意

<u>三维多边形</u>

三维多边形的填色原理与二维情况相同,指令的调用格式也基本相同,这里不再解释。下面是一个简单的例子,请读者自己体会。

【例 2】三维空间中多边形的填色。

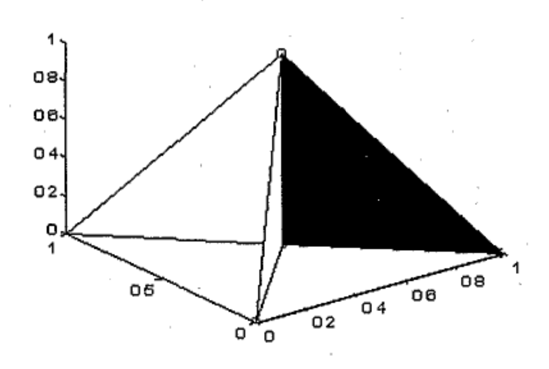


图 5 2-10　三维空间中多边形的填色

```
clf
x1=[0.5,0.5,0]'; y1=[0 5,0 5,0]';z1=[1,0,0]';
x2=[0.5,0 5,1]'; y2=[0.5,0.5,0]';z2=[1,0,0]';
```

```
fill3(x1,y1,z1,'c',x2,y2,z2,'m');hold on
plot3(x1,y1,z1,'ro',x2,y2,z2,'ro');
plot3([0.5,0,0 5]',[0.5,1,0.5]',[1,0,0]','k-')
```

5.3 曲面的表现

MATLAB 表现三维曲面有四种常用形式：网线图(Mesh Plot)；渲染表面图(Shaded Surface Plot)；伪彩图(Pseudocolor Plot)；等高线(Contour)。其中，以前两种尤为基本。MATLAB 还提供控制颜色、光线、视角的指令，从而使三维曲面的表现更加灵活自如。

5.3.1 三维网线图深入

基本的网线图指令 mesh 我们已经在第一节中介绍过了，MATLAB 还提供了另外两个绘制网线图指令：meshc 和 meshz，其中 meshc 是将 mesh 图与等高线图放在一起绘制，meshz 在画图时给出零基准平面。

网线结点 x－y 坐标数据的生成

二元函数 z＝f(x, y)网线图上的“网线结点”(即绘图数据点)满足以下关系：

$$z_{ij} = f(x_{ij}, y_{ij}) \quad i = 1, \cdots, n \quad j = 1, \cdots, m$$

其中，x_{ij}, y_{ij}是网线结点的 x-y 坐标。

当“网线”位置(用 n 维 x 向量和 m 维 y 向量表示)已知，则可用以下指令求得网线结点的 x－y 坐标数组 X、Y，它们都是(m×n)维。

[X, Y]＝meshgrid(x, y)

进而利用数组运算可求得相应的 Z 阵。三元组{X(i, j), Y(i, j), Z(i, j)}代表一个网线结点，三数组{X, Y, Z}表示了全部网线结点。在第 5 1 2 节例 2 有网线结点的图示。

指令 meshgrid 的另一个三元调用格式 [X, Y, Z]＝meshgrid(x, y, z) 将在第 5 3 3 节介绍。

重叠线的消隐和透视

MATLAB 画 mesh 网线图时，(缺省地)对重叠在后面的网线采取了消隐措施。但有时却需要透视效果，特别是当多个图形在同一坐标时。为此，MATLAB 提供了一个消隐开关

hidden on　　　消隐重叠线

hidden off　　　透视重叠线

【例 1】运用透视开关绘制带等位线的网线曲面。

```
meshc(peaks(20));          %带等位线的网线图
colormap([1 0 0])          %红色网线
hidden off                 %透视
```

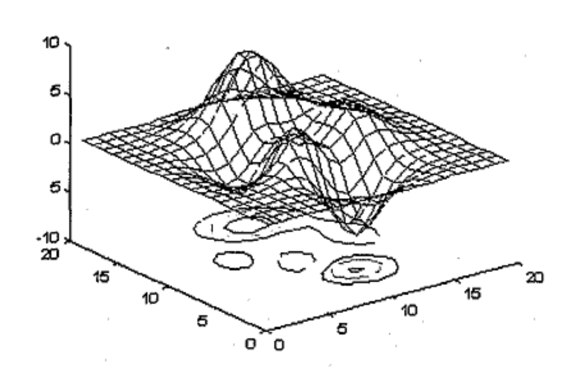


图5 3-1　透视下的带等位线的网线曲面

网线图的裁剪

利用“非数” NaN 的特点，可以对网线图进行裁剪，举例如下。

【例 2】垂帘网线图的剪孔。

```
P=peaks(30);
P(20:23,9:15)=NaN*ones(4,7);          %剪孔
meshz(P);                             %垂帘网线图
colormap([0 0 1])                     %蓝色网线
```

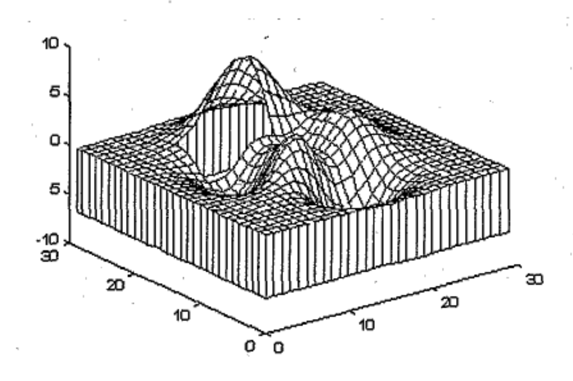


图5 3-2　带剪孔的垂帘网线图

带等高线网线图和垂帘网线图

实际上，读者已在上小节例 1 和例 2 中看到了带等高线网线图和垂帘网线图。因此，本小节只需再作些补充说明即可。

meshc(Z)　　普通网线图，并配有 z=0 平面上的二维等高线图。

meshz(Z)　　普通网线图，并在网线图周边与 z=min(min(Z))平面之间画垂帘线。

这两个指令的调用格式与 mesh 完全相同。

瀑布水线图

瀑布水线图也是网线图的一个变种。绘制瀑布水线图的指令调用格式为：

waterfall(Z)　　　以 Z 阵下标作 x－y 坐标，画瀑布水线图。

waterfall(X,Y,Z)　　以 X,Y 阵元素为 x－y 坐标，画瀑布水线图。

【例 3】瀑布水线图。

```
waterfall(peaks(30));
colormap([0 0 1])
```

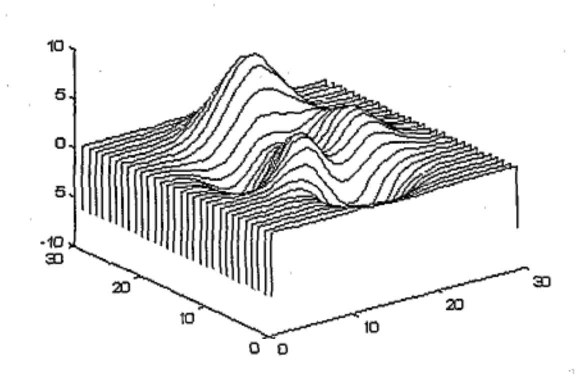


图 5 3-3　瀑布水线图

5.3.2 着色表面图

网线图通过着色网格线表现空间曲面的形状，而着色表面图是通过那张在黑色网格上的着色表面去表现空间曲面的。MATLAB 提供三个绘制着色表面图的指令：surf、surfc 和surfl。其中 surf 是绘制着色表面图的基本指令，surfc 绘制带等高线的着色表面图，而 surfl 可以控制光照效应。

着色表面图的基本指令

surf 指令的用法与 mesh 完全一样，它的调用格式是：

surf(Z,C)　　　以矩阵 Z 的下标为 x-y 坐标，按 C 指定的色彩绘制表面图。

surf(X,Y,Z,C)　以矩阵 X、Y 元素为 x-y 坐标，按 C 指定的色彩绘制表面图。

surf(x,y,Z,C)　分别以向量 x，y 元素为 x-y 坐标，按 C 指定的色彩绘制表面图。

说明：

(1) 在以上格式中，C 是决定表面的每个张面色彩的矩阵（详见稍后关于 shadıng 指令的介绍）。当它缺省时，即默认 C＝Z 。表面的每个张面按相应的 Z 阵元素值上色。

(2) 第三种调用格式中的 x 、y 向量的长度必须分别等于 Z 阵的列维和行维。

(3) 第二种调用格式中的 X、Y 可利用分格函数 [X,Y]＝meshgrıd(x,y) 从第三种调用格式中的 x 、y 获得。

【例 1】椭圆面。

```
x=-1.5:0.3:1.5;y=-1:0.2:1;
[X,Y]=meshgrid(x,y);
Z=sqrt(4-X.^2/9-Y.^2/4);
surfc(X,Y,Z)                %带等高线的着色表面
```

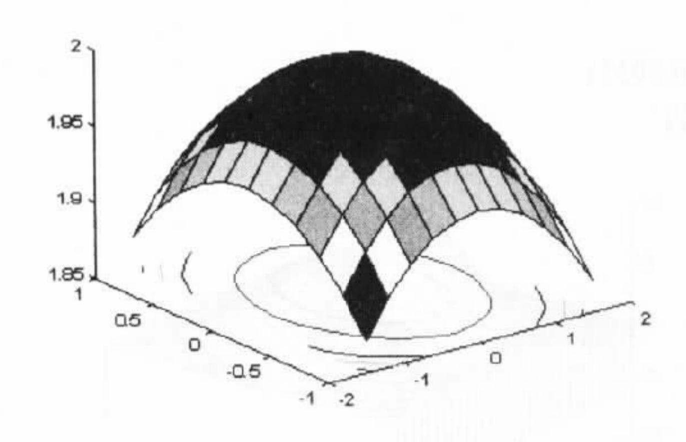


图 5.3-4　椭圆着色表面

带等高线的着色表面图

绘制带等高线表面图的指令 surfc，用法与 meshc 一样，这里不再赘述。图 5.3-4 就是该指令的使用举例。

带光照效果的表面图

关于光照控制方法将在稍后详细介绍，这里给出的例 2 是用 surfl 指令重绘例 1 的椭圆面，以做比较。在缺省情况下，光源被置在从视线角度逆时针旋转 45°处。

【例 2】光照(缺省模式)下重绘椭圆面(图 5.3-5)。

```
surfl(X,Y,Z)
```

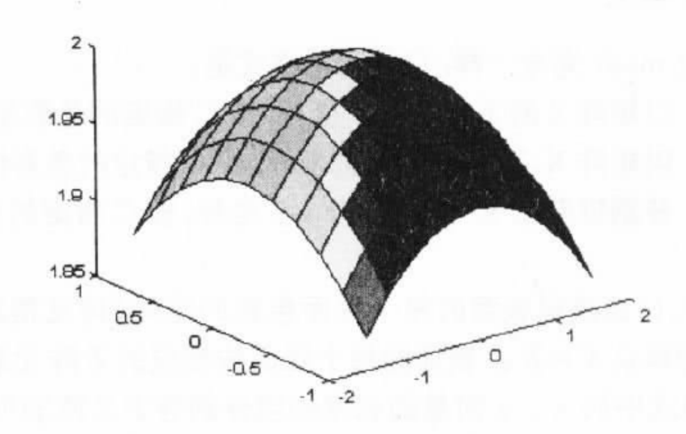


图 5.3-5　光照下的着色椭圆面

5.3.3 二元函数的伪彩图

前面介绍的网线图和表面图都是用三维坐标去表现二元函数 z = f(x, y)的。在这两种图形中，函数值表示为三维空间中的高度。本节将介绍一种用彩色表现函数值的指令——伪彩图(Psudeocolor Plot)。伪彩图指令的调用格式如下：

pcolor(Z)　　以 Z 阵元素的下标为 x-y 坐标画伪彩图。它等价于 surf(0 * Z, Z) 。

pcolor(X, Y, Z)　　以 X、Y 阵的元素为 x-y 坐标画伪彩图。它等价于 surf(X, Y, 0 * Z, Z)。

pcolor(x, y, Z)　　等价于 surf(x, y, 0 * Z, Z) 。

在此所说的“等价”是指，当观察角指令取 view([0 90])时，所看到的“正面俯视”表面图就是伪彩图。Z 阵中的最大值和最小值分别对应色图(Color Map)的第一种和最后一种颜色。Z 阵中的其他元素值分别对应色图的某种颜色，从而不同大小的元素便通过不同的颜色表现了出来。

【例 1】peaks 函数的伪彩图(5.3-6)。

```
Z=peaks(30);
pcolor(Z);
shading flat            %采用 flat 着色模式，使网格线消隐
colormap(cool)          %采用 cool 色图
colorbar('horiz');      %画水平色轴
```

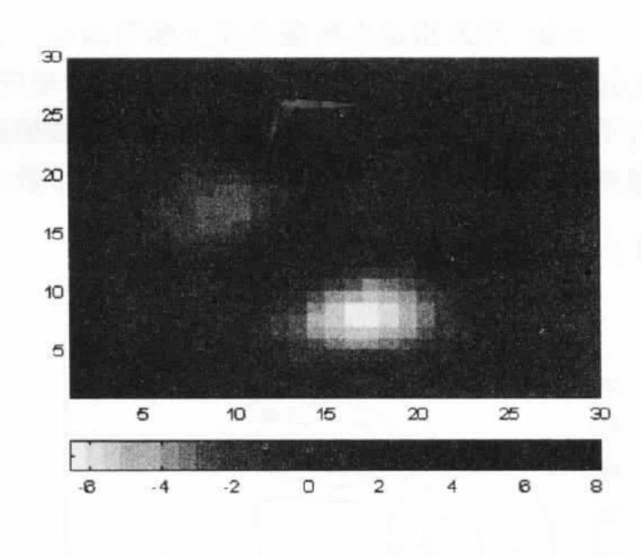


图 5.3-6　伪彩图

说明：

(1) 因为 Z 是(30×30)的数组，所以第二句指令所画伪彩图的横、纵坐标都是从 1 到 30 的正整数刻度。

(2) 本例第五句指令用来画一根水平色轴。该色轴清楚地表明了数值与颜色的对应关系。

5.3.4　等高线

等高线(Contour)是表现三维曲面的第四种形式。MATLAB 支持二维和三维等高线的图形绘制。在绘制时,用户可指定等高线的条数或位置。

在平面上绘制等高线

本节要介绍与绘制平面等高线有关的三个指令:

contour(x, y, Z, n)	绘制 n 条等高线。
contour(x, y, Z, v)	在向量 v 指定的值上绘制等高线。
C= contourc(x, y, Z, n)	计算 n 条等高线的 x-y 坐标数据。
C= contourc(x, y, Z, v)	计算向量 v 所指定的等高线的 x-y 坐标数据。
clabel(C)	给 C 阵所表示的等高线加注高度标识。
clabel(C, v)	给向量 v 所指定的等高线加注高度标识。
clabel(C, 'manual')	借助鼠标给点中的等高线加注高度标识。

说明:

(1) 上述指令中的 x、y 是用来确定 x－y 坐标如何刻度的。它们可以缺省,缺省时,坐标用 Z 阵的下标刻度。

(2) 指令中的 n 是一个标量,用来指定等高线的条数。它可以缺省,缺省时,等高线的条数将自动生成。

(3) 指令中的 v 是一个向量,用来指定在所需高度上画等高线。它也可以缺省。

(4) 由 contourc 所给出的等高线坐标阵 C 是一个行数为 2 的矩阵。它的排列方式是每一条线由该矩阵的一个全行子阵表示:子阵的第一行第一列是等高线的高度值;子阵的第二行第一列是画该等高线的数据点数,子阵的其余列中的每一列都是一个数据点。

【例 1】在二维平面上绘制 peaks 函数的 6 条等高线。

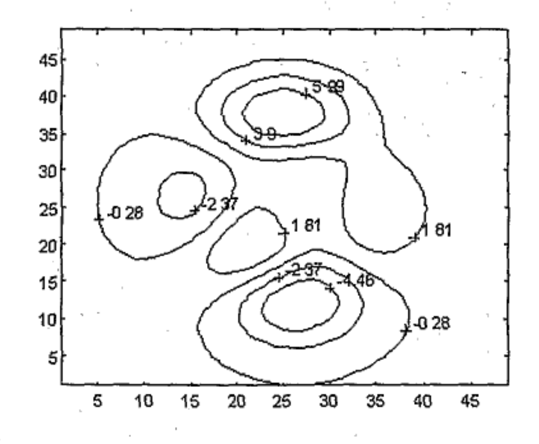


图 5 3-7　等高线及其标注

```
clf
Z=peaks;
contour(Z,6)
C=contourc(Z,6);
clabel(C)
```

说明:等高线指令和伪彩图配合使用可得到很好的效果。

在三维空间中绘制等高线

在三维空间中绘制等高线的指令是 contour3，用法与 contour 类似，这里不再重复。

【例 2】在三维空间中绘制 peaks 函数的等高线。

```
Z=peaks;
contour3(Z,20);
```

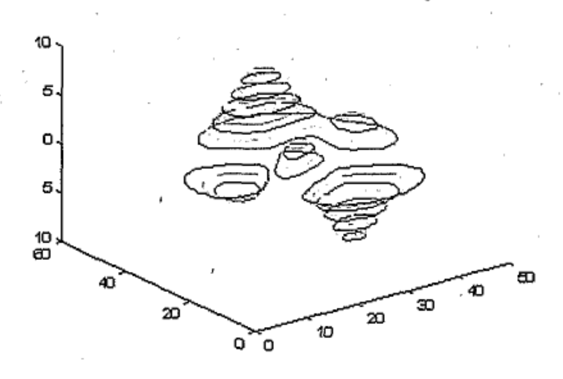


图 5 3-8 三维空间中绘制等高线

5.3.5 矢量场图

矢量场图(又称速度图)是由指令 quiver 实现的。它主要用于描写函数 $z=f(x,y)$ 在点 (x,y) 的梯度大小和方向。该指令的使用示例见第 3 13 4 的例 4，其一般的调用格式为:

quiver(X, Y, DZX, DZY, s, 'LSC')

说明:

(1) DZX、DZY 是该指令调用格式中必不可少的两个输入宗量，它们决定矢量场图中各矢量的大小和方向。它们一般来自求二元函数矩阵 Z 的梯度指令

[DZX, DZY] = gradient(Z, dx, dy)

这里，dx、dy 是 x、y 方向上的计算步长。DZX、DZY 是 $\frac{\partial z}{\partial x}$、$\frac{\partial z}{\partial y}$。

(2) 前两个输入宗量 X、Y 是 Z 阵元素的坐标矩阵，它们与"网线"向量 x、y 的关系如下:

[X, Y] = meshgrid(x, y)

在矢量场图指令 quiver 中，X、Y 也可以用"网线"向量 x、y 替代。

（3）第五个宗量 s 是指定所画箭头的大小的。缺省时，认为 s=1 。

（4）第六个宗量 'LSC'是字符串，指定合法的线型和彩色，有关代码见表 5.1－1 。

5.3.6　柱面和球面

为了方便地生成柱面（Cylinder）和球面（Sphere），MATLAB 提供了两个专门指令 cylinder 和 sphere 。

柱面

柱面采用"母线"旋转生成。"母线"用向量 r 定义，旋转圆周上的分格线条数用 n 定义。于是利用下列指令可得到柱面的 x－y－z 坐标三对组阵 X、Y、Z。

[X, Y, Z]＝cylinder(r, n)

说明：

（1）"母线"向量 r 总被认为是在单位高度里等分刻度上定义的半径向量。

（2）旋转圆周上的分格线条数 n 可以缺省。缺省时，默认 n＝20 。

（3）由该指令所得的 X、Y、Z 可通过网线图指令 mesh 或表面图指令 surf 表现为图形。

【例 1】旋转柱面图（图 5 3-9）。

```
t=0:pi/12:3*pi;
r=abs(exp(-0.25*t) *sin(t));
[X,Y,Z]=cylinder(r,30);
mesh(X,Y,Z)
colormap([1 0 0])
```

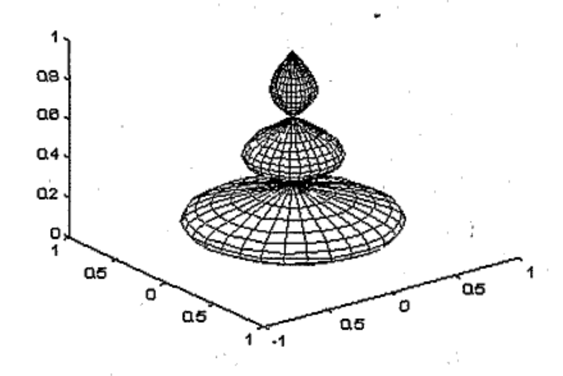


图 5 3-9　旋转柱面图

球面

[X, Y, Z]＝sphere(n)

指令中 X, Y, Z 和 n 的含义均与柱面相同。

【例 2】地球表面气温分布示意(图 5.3-10)。

```
[X,Y,Z]=sphere(30);
T=abs(Z);              %假设气温函数
surf(X,Y,Z,T)
caxis([-max(max(T)) max(max(T))])
colormap(hot)
```

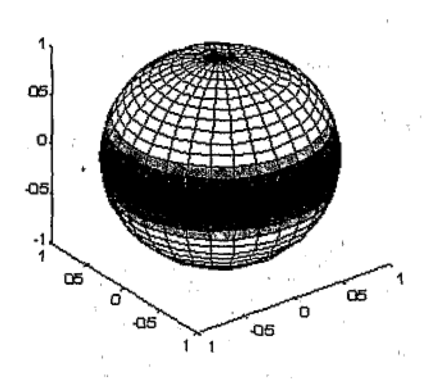


图 5.3-10 地球表面气温分布示意

说明:该例实际上已经涉及到用色彩表现第四维的问题。例中的 T 可以一般地定义为 T=T(x,y,z),然后再用它去控制球表面的色彩。

5.4 四维表现和切片图

对于定义在表面、柱面、球面上的三元函数,都可以采用与第 5.3.6 节例 2 类似的方法处理,从而借助色彩实现四维表达。本节下面介绍,定义在一般 x-y-z 坐标上的四维可视化指令 slice 。

为实现三元函数 $v=f(x,y,z)$ 的可视化表现,MATLAB 提供了一个绘制三维物体切片图指令及与之配合使用的三维网格坐标生成指令。

[X,Y,Z]=meshgrid(x,y,z)　　三维网格坐标的生成。

slice(X,Y,Z,V,xi,yi,zi,n)　　绘制三维物体切片图。

其中:

x、y、z　　决定"网线"位置,分别是(1×n)、(1×m)、(1×p)向量。

X、Y、Z　　三维网格坐标,它们都是(n×m)×p 维数组。

V　　在网线结点上的三元函数值数组,维数也为(n×m)×p 。

xi、yi、zi　　分别决定垂直于 x、y、z 轴切面位置向量。它们的维数可以各自不同。当取 0 维空阵"[ ]"时,表示没有切面存在。

切片上函数值的大小用色轴上对应的颜色表示。V 数组中的最大有限值和最小有限值定义了色轴的范围。又由于切片位置可任意设置,因此 slice 通过三维坐标点上的色彩变化把图

形的表现能力扩展到了四维。

【例 1】函数 $\nu = xe^{-(x^2+y^2+z^2)}$ 的四维表现。

```
x=-2:0.1:2;y=-2:0.25:2;z=-2:0.25:2;n=length(x);
[X,Y,Z]=meshgrid(x,y,z);
V=X.*exp(-X.^2-Y.^2-Z.^2);
xi=[-0.7,0.7];yi=0.5;zi=-0.5;                    %选择切面位置
slice(X,Y,Z,V,xi,yi,zi,n);                       %画切面图
xlabel('x');ylabel('y');zlabel('z');hold on
colorbar('horiz')                                %第四维坐标,即色轴
view([-30 45])                                   %观察角控制
```

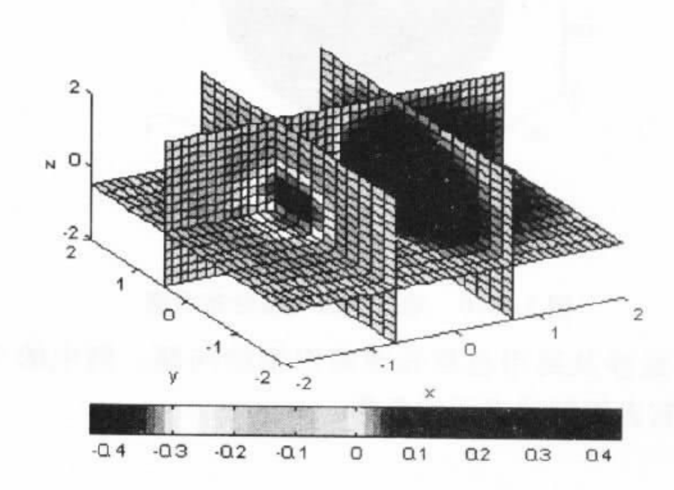


图 5.4-1 四维表现

说明:

(1) 解析计算表明:

在 x=0.7071, y=0, z=0 时, max(ν)=0.4289。

在 x=-0.7071, y=0, z=0 时, max(ν)=-0.4289。

(2) 色轴在图的最下方。它标注着颜色和数值的对应关系。

(3) 十分遗憾,读者无法直接从本书见到第四维色彩的美妙表现。考虑到黑白印刷,所以在本例中采用了缺省色图。这是因为它用黑白印刷时,灰度层次较明显。假如读者能在MATLAB 环境中实践,那就可以体验和欣赏到四维表现的奇妙。

5.5 图形的标注

MATLAB 可以在画出的图形上加各种标注及文字说明,以丰富图形的表现力。在中文 Windows 操作环境下,还可以用汉字进行标注与文字说明。但是需要注意,用于标记文字的单引号应该是英文而不是中文的单引号。

5.5.1 图名和坐标轴名的标注

MATLAB 为定义图形名称和坐标轴名称提供了专门的标注指令。在 MATLAB 4 0 及以后的版中，加注的内容不仅可以是外文字符，而且还可以是中文字符。具体如下：

tıtle('Strıng') 在图形的顶端加注文字作为图名。

tıtle('Strıng', 'Property' , Propertyvalue,) 定义图名所用字体、大小、标注角度。

xlabel('Strıng') 在当前图形的 x-轴旁边加入文字内容。

xlabel('Strıng', 'Property' , Propertyvalue,) 定义轴名所用字体、大小、标注角度。

legend('strıng1','strıng2',) 对当前图进行图例标注。

说明：

(1) 关于 tıtle 的使用示例见第 5 6 2 节例 1，第 5 6 3 节例 1 等。

(2) 给 y、z 轴加注的指令是 ylabel、zlabel，它们的调用格式与 xlabel 完全一样(见第 5.6 2 节的例 1)。

(3) 加注内容 Strıng 可以是中文。但在 MATLAB 3 0 版及 3 5 版中，不允许用中文标注。

(4) 第二种调用格式涉及图形对象的"句柄"操作，在此不作介绍，有兴趣读者可参看本书第 5 11 3 节和 MATLAB 的在线帮助。MATLAB 4 0 版没有这种调用格式。

(5) 关于 legend 的例子参见第 3 9 3 节例 5 中图 3 9-1 和第 3 13-3 节例 1 中图3 13-1。如果用户想改变图例说明框的位置，可以按住鼠标把它移动到合适的地方。

5.5.2 所画图形的标注

MATLAB 还提供对所绘图形的文字标注功能：指令 text，在图形中指定的点上加注文字；gtext 指令，先利用鼠标定位，再在此位置加注文字，该指令不支持三维图形。

text

text(x, y, z, 'strıng') 在点(x, y, z)上标注文字 strıng 。

text('PropertyName', PropertyValue,) 对标注的文字属性加以定义。

说明：

(1) 如果 x、y、z 是向量，则在这三个向量定义的每一点上标注文字 strıng 。如果 strıng 也是由字符串组成的字符列向量，则在每一点上标注对应的字符。

(2) 在第一种调用格式中，若没有 z，那就是在二维平面图形上加注(见第 5 1 1 例 5)。

(3) 关于第二种调用格式的介绍，请见第 5 11 节例 11。

gtext

gtext('strıng') 在鼠标指定位置上加注。

说明：当这个指令输入后，会在当前图形上出现一个十字叉，等待用户选定标注位置。移动鼠标到所需位置按下鼠标按钮，MATLAB 就在选定位置标上文字。

5.5.3 分格线

grid on　　打开分格线绘制开关，使此后绘制的图形都带分格线。
grid off　　关闭分格线绘制开关，使此后绘制的图形都不带分格线。
grid　　实现分格线绘制与否的切换。
该指令对二维或三维图形都适用，且使用频度较高，使用方法简单，如第 5 1 1 例 5 。

5.6 图形的控制与表现

本节将要介绍怎样来控制和增强由前面的绘图函数所产生的图形的表现力。由于 MATLAB 专门提供了一组指令来对图形的外表进行控制，这样用户在工作时就显得得心应手，专心于自己的工作而不必为一些琐事操心。另外，还有很多图形控制与表现的函数(如：颜色的控制、图形对象的句柄操作、图形的动态表现等)将在后面专门介绍。

5.6.1 图形的窗口创建和控制

通常，每当在 MATLAB 指令窗中的第一个绘图指令运行后，就自动创建一个名为"Figure No 1"的图形窗口。此后，它就被作为当前窗口。这意味着，从此之后的绘图指令所画的图形都将出现在这个"Figure No 1"的图形窗口，或把先前的图形覆盖掉，或叠加在先前的图形上。在以上的运作过程中，用户并没有直接和 figure 指令打交道，对图形窗口的管理是按默认方式进行的。

若用户在保留原先图形的同时希望绘制一幅新图形，那么就需要使用命令来对图形窗口进行操作。图形窗口指令的调用格式如下：

figure　　每调用一次就打开一个新的图形窗口。
figure(n)　　创建或打开第 n 个图形窗口，使之成为当前窗口。

说明：

(1) 欲清除当前图形窗口中的所有内容，可使用 clf (或 clg)指令。该指令的详细用法，请用 help 功能在线查询。

(2) 若只清除当前图形窗口中所画的图形，而保留其坐标，则可用 cla 指令。

5.6.2 子图形的创建和控制

在一个图形(Figure)窗口里，可以画多个有独立坐标系统的子图形，指令如下：

subplot(m, n, p)　　将图形窗分割成 m×n 个子图，并选择第 p 个子图为当前图形。

说明：

(1) 子窗口的序号按行由上往下、按列自左至右编号。

(2) 如果不用指令 clf 清除，以后图形将被绘制在子图形窗口中。

【例 1】子窗口及不同坐标系的对照。

```
clf;[X,Y,Z]=peaks(20);
colormap([1,0,0])
subplot(2,2,1);mesh(Z);axis('ij');grid
title('矩阵坐标');xlabel('J 轴');ylabel('I 轴')
subplot(2,2,2);mesh(Z),axis('xy');grid
title('下标刻度笛卡尔坐标');xlabel('X 轴 ');ylabel('Y 轴');
subplot(2,2,4);mesh(X,Y,Z);axis('xy');grid
title('X-Y 刻度笛卡尔坐标');xlabel('X 轴 ');ylabel('Y 轴');
```

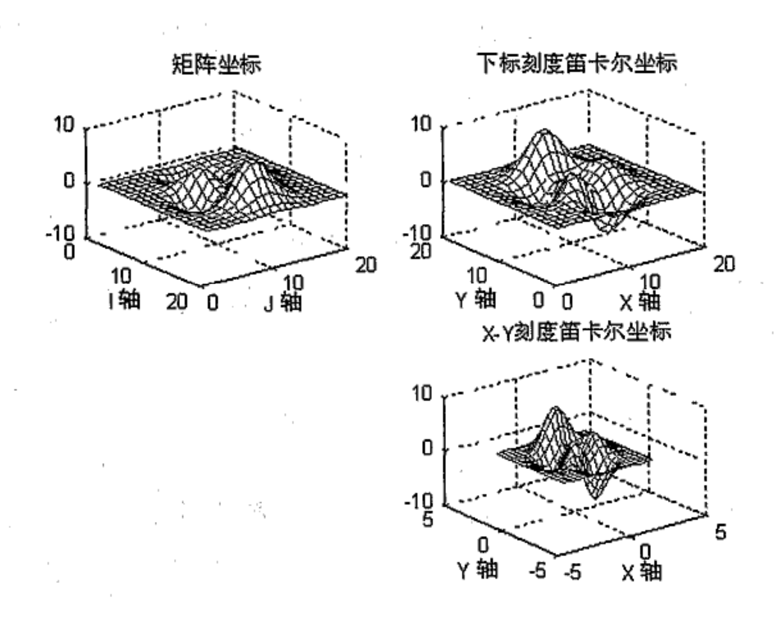


图 5 6-1 子窗口及不同坐标系的对照

说明:关于不同坐标的含义,请看第 5 6 4 节和第 5.1 2 、5 3 1 节。

5.6.3 图形的重叠绘制

如果用户想在某图形窗口中同时绘制多种不同属性的图形对象,最简单的方法是使用命令 hold 。该指令有下面三种调用格式:

hold on 保留当前图形和它的轴,使此后图形叠放在当前图形上。

hold off 返回 MATLAB 的缺省状态:此后图形指令运作将“抹掉”当前窗中的旧图形,然后画上新图形。

hold 在上述两个状态间切换。

【例 1】已知两组数据，用最小二乘法求拟合直线 y = ax + b 。

```
x0=[1.1, 1.2,1 3,1 4,1 5,1.6,1.7,1.8,1.9,2 0];
y0=[123, 130,141,155,169,171,183,190,205,210];
plot(x0,y0,'b*');              %画被拟合数据
hold on
A=[x0',ones(size(x0'))];B=y0';
x=A\B;                         %求最小二乘拟合系数
a=x(1); b=x(2);y=a*x0+b;
plot(x0,y,'r');                %画回归直线
title('用最小二乘法拟合数据');
string=['拟合直线 y = ',num2str(x(1)),' * x + ',num2str(x(2))];
text(1.3,200,string);
hold off
```

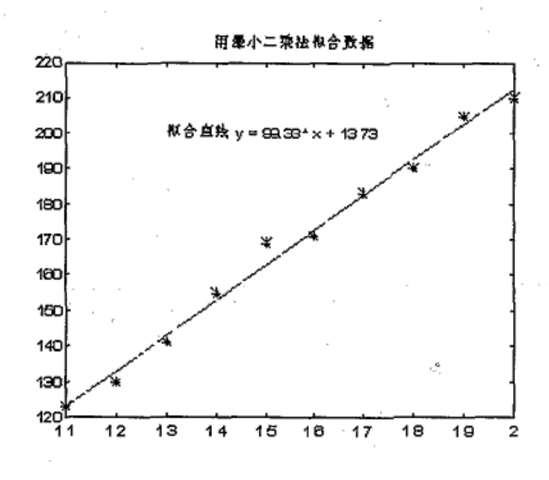


图 5 6-2 最小二乘法求拟合

5.6.4 坐标轴的控制

在缺省情况下，MATLAB 图形坐标轴的表现状态是：“自动(Auto)”对坐标轴刻度、“显示(on)”坐标轴、采用“直角(xy)”坐标。在这种工作状态下，用户不需要对图形坐标进行任何干预，坐标刻度范围将根据绘图指令中向量或矩阵中的元素自动决定。在很多情况下，这种缺省工作状态给绘图带来很大的方便。

图形是多种多样的，统一坐标模式不可能总是最有效地表现出所绘图形的特征。于是，MATLAB 设计了操纵坐标性质的 axis 指令。它的主要调用格式如下：

axis([xmin, xmax, ymin, ymax])	指定二维图形 x-轴和 y-轴的刻度范围。
axis([xmin, ymax, ymin, ymax, zmin, zmax])	指定三维图形三个轴的刻度范围。
axis('auto')	返回坐标轴的缺省状态。

axıs(axıs) 保持刻度范围不变。

axıs('ıȷ') 以“矩阵(ıȷ)”坐标轴表现图形。

axıs('xy') 使坐标轴回到(缺省状态的)笛卡尔坐标系。

axıs('off') 使坐标轴消隐。

axıs('on') 返回(缺省的)坐标轴显现状态。

axıs('equal') 使各坐标轴刻度增量相同。

axıs('square') 使各坐标轴长度相同(但刻度增量未必相同)。

axıs('normal') 使上述两个指令作用失效。

说明:

(1) 当前图形窗口坐标轴的工作状态可用以下指令进行查询.

[S1,S2,S3]=axıs('state')

S1 若是 auto,则坐标轴自动刻度;若是 manual,则坐标轴人工刻度。

S2 若是 on,则显示坐标轴;若是 off,则消隐坐标轴。

S3 若是 xy,则采用笛卡尔坐标;若是 ıȷ,则采用“矩阵”坐标。

(2) 所谓笛卡尔坐标是指:坐标系的原点位于左下角;x 轴水平放置,自左至右刻度;y 轴垂直设置,由下往上刻度。所谓“矩阵”坐标是:坐标系的原点位于左上角;ı 轴垂直设置,由上而下刻度;ȷ 轴水平放置,由左到右刻度。

(3) 关于不同坐标系对照的举例,请看第 5 6 2 节的例 1。

(4) axıs('square')指令使用示例见第 5 1 1 节例 4 。

5.6.5 视角的设置

MATLAB 允许用户通过观察函数 vıew 定义所需的观察点。观察函数最常用的格式是:

vıew(az,el) 球坐标中设置观察点。az 是方位角,el 是俯视角。

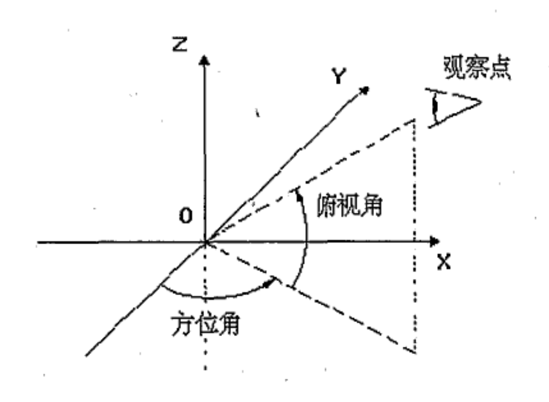


图 5 6-3 观察点的定位

说明:

(1) 方位角(Azımuth)和俯视角(Elevatıon)的计值基准面和方向如图 5 6-3 所示。角度的

单位是“度”。进行二维观察时的缺省值为：az = 0，el = 90；进行三维观察时的缺省值为：az = −37.5，el = 30。

(2) 观察点也可以用直角坐标形式指定，格式为：view([x, y, z])。

(3) 观察点还可以直接用观察变换矩阵 T 设定，格式为：view(T)。

(4) 当前观察点位置或变换矩阵可用以下指令获得：

[az, el] = view　　给出当前观察方位角和俯视角。

T = view　　给出当前观察变换矩阵。

【例 1】peaks 函数的四种不同视图。

```
z=peaks(40);
subplot(2,2,1);
mesh(z),view(-37.5,30),title('方位角=-37.5 俯角=30');
subplot(2,2,2),mesh(z),view(-7,80),title('方位角=-7 俯角=80');
subplot(2,2,3),mesh(z),view(-90,0),title('方位角=-90 俯角=0');
subplot(2,2,4),mesh(z),view(-7,-10),title('方位角=-7 俯角=-10');
```

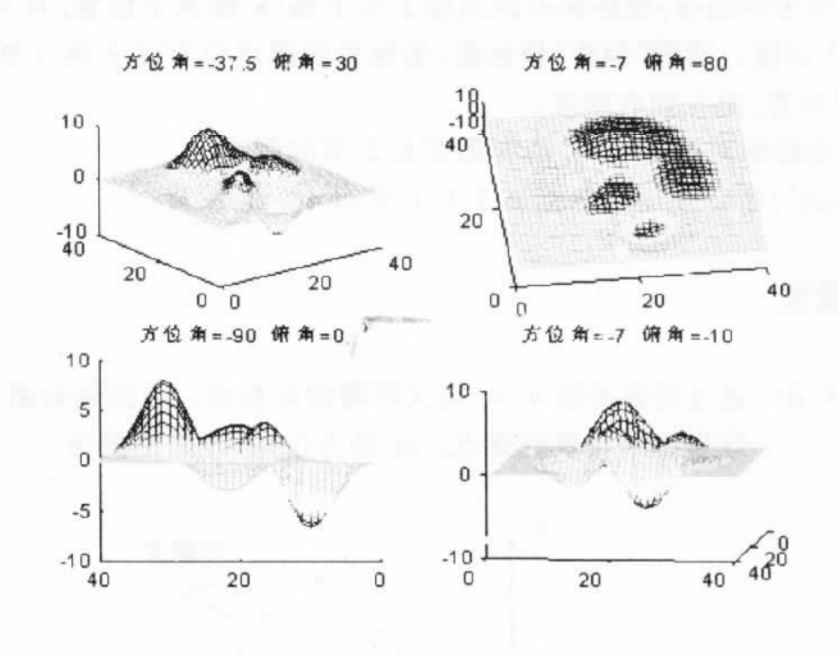


图 5.6-4　不同观察点下的图形

5.6.6　光照控制

surfl(x, y, Z, S, K)　　产生带光照的明暗处理三维表面图

说明：

(1) 与 surf 指令一样，x、y、Z 定义三维表面，x、y 可以缺省。

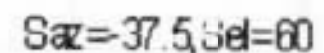
(2) S 确定光源位置。它可以由直角坐标定义，即 S = [sx, sy, sz]；也可以由球坐标中的方位角和俯视角定义，即 S = [az, el]。S 可以缺省。缺省时，光源的方位角在观察点逆时针方

向 45° 处。

(3) K 是确定用光模式的向量,K = [ka, kd, ks, spread] 。在此,ka 是背景光(ambient)份额;kd 为漫射光(diffuse)份额、ks 为定向光(specular)份额、spread 是扩散系数。K 一般缺省。

【例 1】peaks 函数在控制光照下的三维表面(图 5.6-5)。

```
clf
surfl(peaks,[-37.5,60]);colormap(gray);shading interp
title('Saz=-37.5,Sel=60')
```



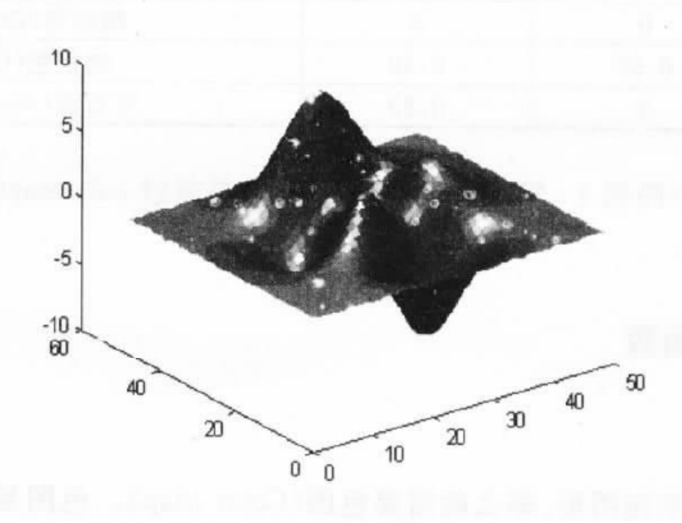


图 5.6-5 带光照的明暗处理表面

5.7 色彩的控制与表现

颜色在图形的表现中起着非常重要的作用。它不仅给人以美的享受,而且从数学上讲,它给几何坐标多增加了一维表现能力(见第 5.3 和 5.4 节)。本章将更深入地介绍由 mesh 、surf、pcolor 等指令所产生图形色彩和渲染的控制与表现。

5.7.1 色彩的调制

在 MATLAB 中,一种色彩用一个三元行数组表示。该三元数组为 [R G B] ,元素 R、G、B 的取值必须在 0 和 1 之间;元素 R、G、B 的数值大小分别表示红、绿、蓝基色的相对亮度。通过对 R、G、B 大小的不同设置,就可"调制"出不同的颜色。表 5.7-1 给出了一些常用颜色所对应的三元数组的具体取值。这三元数组通过如下指令而变成当前绘图用色:

colormap([R G B])　　　　设置当前绘图用色。

表 5.7-1 典型调和色

基	色		调和色
红(R)	绿(G)	蓝(B)	
1	1	1	白色(White)
0.5	0.5	0.5	灰色(Gray)
0	0	0	黑色(Black)
1	0	0	红色(Red)
0	1	0	绿色(Green)
0	0	1	蓝色(Blue)
1	1	0	黄色(Yellow)
1	0	1	品红色(Magenta)
0	1	1	青色(Cyan)
0.5	0	0	暗红色(Drak Red)
1	0.62	0.40	纯铜色(Copper)
0.49	1	0.83	宝石蓝(Aquamarine)

比如,第 5.1.2 节中的例 1 、第 5.3.6 节的例 1 就是通过 colormap([1 0 0]) 指令使绘制网线图时采用红色进行。

5.7.2 色图和色图函数

色图

若希望用多种彩色绘制图形,那么就需要色图(Color Map)。色图是一个(m×3)的矩阵:它的元素在 [0,1] 闭区间中取值;它的每一行就代表一种颜色。色图既可以通过列表中元素的直接赋值定义,也可以按某种数学规律生成:

与三元数组一样,色图也是通过以下方式变成绘图用色的

colormap(Cm)　　　　生成色图 Cm 所指定的色彩对照表。

【例 1】用给定的(m×3)矩阵,控制图形窗的绘图色彩(图 5.7-1)。

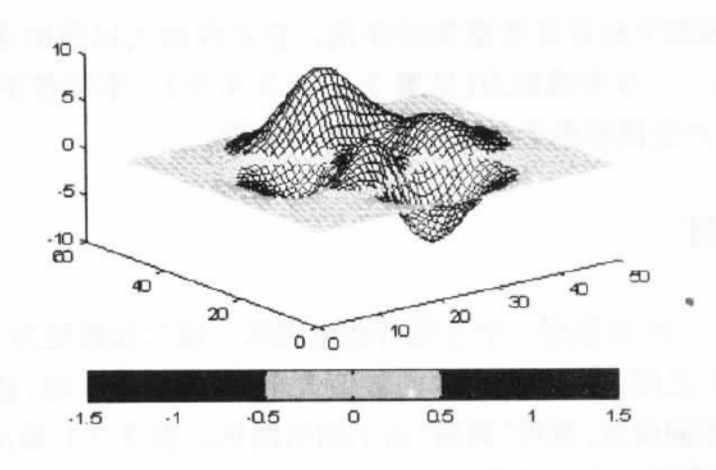


图 5.7-1 用(3×3)矩阵指定的色彩绘图

```
Cm=[1 0 0;0 1 0;0 0 1]; %指定红、蓝、绿三色图
Z=peaks(50);            %取待画函数值
mesh(Z);                %画Z阵的网线图,并默认由Z阵的值决定各网点的着色
colormap(Cm),           %据色图Cm为当前图形窗配置颜色对照表
caxis([-1,1]);          %设置色轴刻度
colorbar('horiz')       %在当前图形窗中,绘制水平放置的色轴
```

说明:在屏幕上三种颜色清楚表明色图与数据矩阵之间的关系,但黑白打印无法表现。

色图函数

色图也可以由函数生成。但不管怎样生成,请用户切记:色图必须是(m×3)矩阵,其元素值必须在[0 , 1]区间内。

在MATLAB中,有一组能生成各种典型颜色对照表的色图函数(见表5 7-2)。

表5.7-2 常用色图函数

色图函数名	色 图 性 质
bone	蓝色调灰度
cool	青和品红浓淡色
copper	线性变化纯铜色调
flag	红-白-蓝-黑交错色
gray	线性灰度
hot	黑-红-黄-白色
hsv	带饱和值的色图
jethsv	色图的一种变体
pink	淡粉红色图
prism	光谱色图

假如对绘图的颜色对照表不作专门设置,那么默认的缺省色图是hsv 。它的颜色排列次序为:红、黄、绿、青、蓝、品红、红。从这个意义讲,hsv色图特别适合于显示周期函数图形。它有下面两种使用格式:

hsv　　　　色图矩阵维数为(64×3)。

hsv(m)　　色图矩阵维数为(m×3)。

其他色图函数的使用方法与hsv相同。

【例2】随机产生色图,并显示其对应的色彩(图5 7-2)。

```
clf;
rand('seed',0)
Cm=rand(4,3);
colormap(Cm);
pcolor([1:5;1:5]');
axis('off')
```

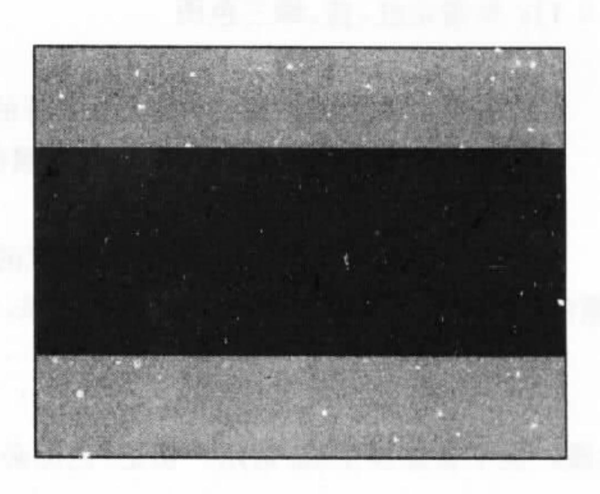


图 5.7-2 本例随机色图所对应的色彩

【例 3】产生一个 10×3 的 gray 色图。

```
map=gray(10)
map =
         0         0         0
    0.1111    0.1111    0.1111
    0.2222    0.2222    0.2222
    0.3333    0.3333    0.3333
    0.4444    0.4444    0.4444
    0.5556    0.5556    0.5556
    0.6667    0.6667    0.6667
    0.7778    0.7778    0.7778
    0.8889    0.8889    0.8889
    1.0000    1.0000    1.0000
```

说明：色图矩阵中各列相等(这实际上说明了灰色是如何产生的)。map 的第一行为黑色，最后一行为白色。

【例 4】图示 hsv 色图的构作和物理意义(图 5.7-3)。

```
rgbplot(hsv)           %用红绿蓝三色分别画 hsv 矩阵的三根列向量
colormap(hsv);         %把 hsv 指定为当前图形窗的颜色对照表
axis([1,64,0,1])       %因 hsv 行维为 64 和元素值在[0,1]间，故对轴做此设定
colorbar('horiz')      %画水平色轴
```

说明：

(1) 运行 hsv 将给出一个(64×3)的矩阵。rgbplot(hsv) 指令的作用是，把 hsv 色图矩阵的三个列向量分别用红、绿、蓝线条绘制在图 5.7-3 的直角坐标上。该图最下方的色轴是由 colorbar('horiz') 产生的。(在黑白打印时，分辨不出红、绿、蓝线。)

(2) 图 5.7-3 的横坐标是 hsv 色图矩阵的“行下标”。纵坐标表示基色的亮度，0 最暗、1

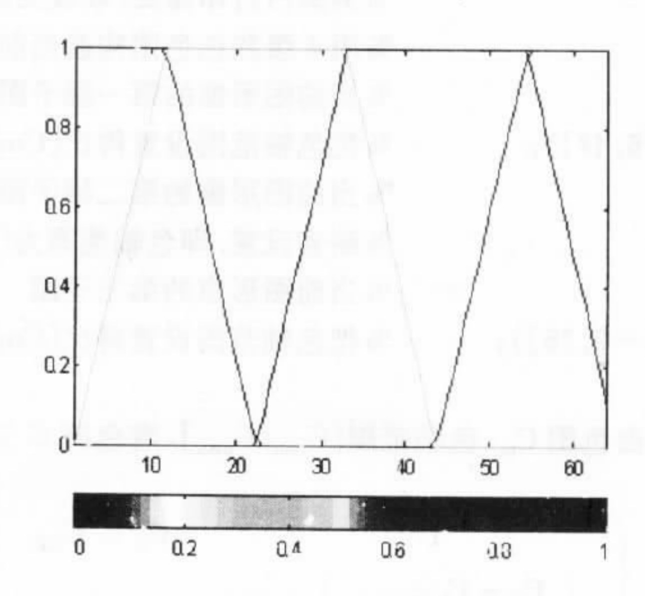


图 5.7-3 hsv 色图的生成

最亮。图中的红线、绿线、蓝线分别表示在不同“行下标”上红色、绿色、蓝色的亮度值。而色轴上的颜色正是由相应横坐标上的三个基色调和而成。

(3) 矩阵的操作同样适合于色图矩阵。如表 5.7-2 中 pink 色图矩阵就是由以下运算而得：pink = sqrt (2/3 * gray + 1/3 * hot)

(4) MATLAB 中的亮度函数 brighten 就是通过对色图矩阵的运算实现亮度调节的。

5.7.3 伪彩着色机理和色轴的设置

在 MATLAB 中有四个基本图形指令(image 、mesh 、pcolor 、surf)是运用伪彩机理给图形着色的。下面以 pcolor 图形函数的一段相关指令为例，说明这种伪彩着色机理。

【例 1】伪彩着色机理图 5.7-4 的生成指令。

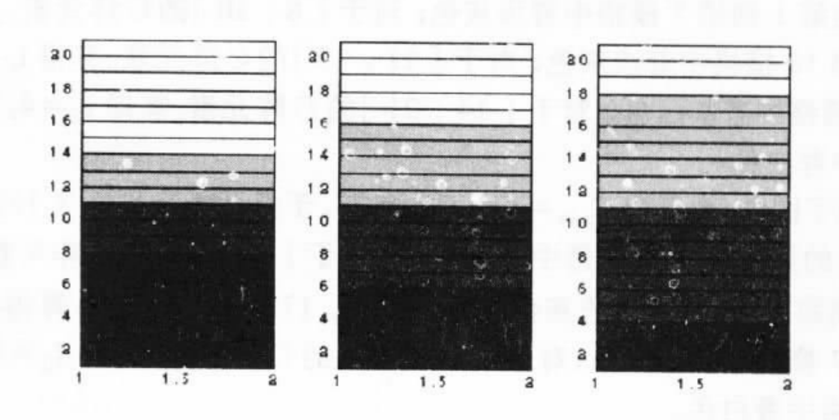


图 5.7-4 伪彩着色机理的图示

```
C=[1:21;1:21]';                    %(2×21)维的着色矩阵
Cm=gray(6);                        %生成 6 级线性灰色色图
```

```
Cm([1 2],:)=[];                    %为黑白打印需要,取较亮的4级为工作色图。
colormap(Cm);                      %用4级灰色色图构造当前图形窗的颜色对照表。
subplot(1,3,1);                    %当前图形窗的第一幅子图
pcolor(C);caxis([5,17]);           %把色轴范围设置得比[Cmin,Cmax]小
subplot(1,3,2);                    %当前图形窗的第二幅子图
pcolor(C),                         %缺省设置,即色轴范围为[Cmin,Cmax]
subplot(1,3,3);                    %当前图形窗的第三子图
pcolor(C);caxis([-3,25]);          %把色轴范围设置得比[Cmin,Cmax]大
```

说明:

(1) 伪彩图的彩色是由色图 C_m、色轴范围$[C_{min}, C_{max}]$、着色阵 C 三者共同决定的。着色规则是:

$$Ic = \begin{cases} 1 & C_{ij} < C_{min} \\ fix\left(\dfrac{C_{ij} - C_{min}}{C_{max} - C_{min}} \times m\right) = 1 & C_{min} \leqslant C_{ij} < C_{max} \\ m & C_{ij} \geqslant C_{max} \end{cases} \quad (1)$$

在此,m 是色图 C_m 的行维;C_{ij}是着色阵 C 的元素;Ic 是 colormap(C_m) 所生成颜色对照表的色彩序号。

(2) 在本例中,色图的行维为 4 。颜色对照表中有四种不同的灰度:深灰(色彩序号为 1);浅灰(色彩序号为 2);灰白(色彩序号为 3);白(色彩序号为 4)。

(3) 图 5.7-4 中的第二幅子图采用缺省色轴设置, 即 $C_{min} = min(min(C)) = 1$, $C_{max} = max(max(C)) = 21$。

对于在 [1 , 5] 之内的 C 色阵元素,据式(1)可以算得 $I_C = 1$,因此用深灰色; 对于在 [6 , 10] 之内的 C 色阵元素,据式(1)可以算得 $I_C = 2$,因此用浅灰色; 对于在 [11 , 15] 之内的 C 色阵元素,据式(1)可以算得 $I_C = 3$,因此用灰白色; 对于在 [16 , 21] 之内的 C 色阵元素,据式(1)可以算得 $I_C = 4$,因此用白色。

(4) 在第一幅子图中,由于取 $C_{min} = 5, C_{max} = 17$,于是, 对于 [1, 7]的 C 阵元素,算得 $I_C = 1$,因此在图上的第 1 到第 7 横格中着深灰色; 对于 [8 , 10]的 C 阵元素,算得 $I_C = 2$,因此在图上的第 8 到第 10 横格中着浅灰色; 对于 [11 , 13]的 C 阵元素,算得 $I_C = 3$,因此在图上的第 11 到第 13 横格中着灰白色; 对于 [14 , 21]的 C 阵元素,算得 $I_C = 4$,因此在图上的第 14 到第 20 横格中着白色。

(5) 在第三幅子图中,由于取 $C_{min} = -3, C_{max} = 25$,于是, 对于 [1, 3]的 C 阵元素,算得 $I_C = 1$,因此在图上的第 1 到第 3 横格中着深灰色; 对于 [4 , 10]的 C 阵元素,算得 $I_C = 2$,因此在图上的第 4 到第 10 横格中着浅灰色; 对于 [11 , 17]的 C 阵元素,算得 $I_C = 3$,因此在图上的第 11 到第 17 横格中着灰白色;对于 [18 , 21]的 C 阵元素,算得 $I_C = 4$,因此在图上的第 18 到第 20 横格中着白色。

(6) 更一般地说,当色轴范围小于着色阵的元素取值范围时,在所得的伪彩着色图上,将更多的使用颜色对照表两端的颜色;反之,当色轴范围大于着色阵的元素取值范围时,在所得的伪彩着色图上,将较少使用(甚至不用)颜色对照表两端的颜色。

(7) colormap 的作用是把色图转化为物理(由计算机硬件体现)的颜色对照表。它对"fig-

ure"图形窗进行的设定，因此在该窗口里所有子图都使用同一个颜色对照表。该指令有以下几种调用格式：

colormap(map)	用 map 色图矩阵设置当前图形的色图。
colormap('default')	采用缺省色图 hsv 。
map = colormap	获取当前图形所使用的色图矩阵。

关于该指令的使用实例在本章前面几节中已经很多，如第 5.1.2 节的例 1；第 5.3.1 节的例 2、例 3；第 5.3.3 节的例 1；第 5.3.6 节的例 1、例 2；第 5.6.2 节的例 1 。因此，这里不再另举算例。用户可以通过改变前面算例中的色图矩阵，去体验该指令的作用。

(8) 色轴范围设置指令 caxis 是对"axis"轴对象进行设定，因此不同的子图可以设置不同的色轴范围；caxis 对其后的绘图函数没有作用。第 5.7.2 节的例 1 就是使用 caxis 压缩中间色彩的。

5.7.4 色彩的渲染

函数 shading 设定由 mesh 、surf 、pcolor 、fill 和 fill3 所创建图形的渲染方式。函数的调用有三种方式：

(1) shading flat

将图形色彩渲染为平坦状态，即据线段两端(或表面小方块四角)的值决定网格线段(或小方块面)取一对应的颜色。

(2) shading interp

将图形色彩渲染为插补状态，即网格线段(或表面)上每个点的颜色是据线段端点(或小方块面四角)值通过双线性插补所算得的值决定的。因此线段(或小方块面)每个点的颜色是渐变的。

(3) shading faceted

缺省方式。在没有对色彩渲染方式专门加以定义时，就采用这种渲染方式。它是在 shading flat 的基础上再加上黑色网格线。这是最有效的渲染方式。使用缺省方式画图的实例见第 5.3.2 节的例 1 和例 2 。

【例 1】使用 flat 模式画图 5.7-5 的指令例示。

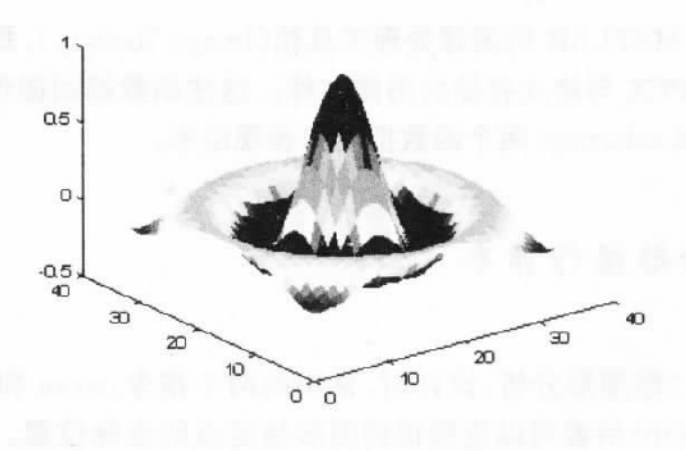


图 5.7-5 使用 flat 渲染模式画图

```
clf;x=-8:0.5:8;y=x;[X,Y]=meshgrid(x,y);
R=sqrt(X.^2+Y.^2)+eps;Z=sin(R)./R;
```

```
surf(Z);colormap(hsv);
shading flat;
```

5.7.5 图像表现函数

image(X) 表现图像的矩阵。

colormap(map) 为表现该图像所给的特定色图。

注意:image 是专供绘制图像(包括照片、绘画等)用的。为了得到真实的图像,必须将图像矩阵和相应的色图同时使用。

【例 1】图像的表现(图 5.7-6)。

```
load gatlin          % 提取在 MATLAB \ DEMOS 子目录下的一组图像数据
image(X)
colormap(map)
axis equal;axis('off') % 保持图像比例和消隐坐标轴
```

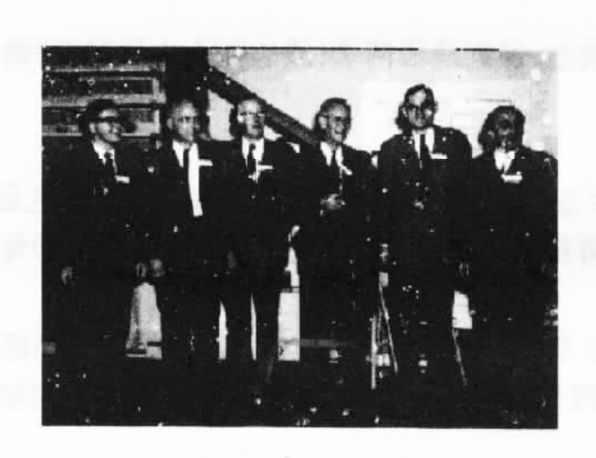


图 5.7-6 六位国际著名数值计算专家在 1964 年数值代数会议期间合影
(左起 Wilkinson, Givens, Forsythe, Housholder, Henrici, Bauer)

说明:假如用户有 MATLAB 的图像处理工具箱(Image Toolbox),那么可以用该工具箱中的函数去读取以 GIF、PCX 等格式存储的图像文件。这些函数返回图像矩阵 X 和色图 map,于是就可以用 image 和 colormap 两个函数把图像表现出来。

5.8 两个特殊图形操作指令

本节介绍在利用二维图形分析、设计时,常用的两个指令:zoom 和 ginput 。前者可以对二维图形进行放大和缩小;后者可以直接得到图形指定点的坐标位置。这两个指令的配合使用会给分析、设计带来很大方便。

5.8.1 变焦指令 zoom

在科学分析和工程设计中，常常需要对计算所得的曲线、图形的"细部"进行观察。MATLAB考虑到这种需求，设计了变焦指令。它在 MATLAB 指令窗中的调用格式是：

zoom on	使当前图形窗处于变焦工作状态。
zoom off	在当前图形窗对象大小不变前提下，恢复普通状态．即非变焦工作状态。
zoom out	使当前图形窗恢复到变焦前的原来大小。
zoom	在 zoom on 和 zoom off 两个状态间切换。

说明：在变焦工作状态下，变焦的实现是靠以下两种鼠标操作进行的：

(1) 点击式操作

移动鼠标到图形上所需放大观察的地方，单击鼠标左键，则鼠标周围的局部将被放大一倍。此后，若再点击左键，则将再次放大；若点击右键，那么图形将恢复到一次放大前的大小。

(2) 拖拉式操作

把鼠标移到需要放大部分的左上方；按住鼠标左键并使鼠标往右下方移动，会出现一个"取景框"；使"取景框"包容待放大部分，放开左键便得局部放大图。

5.8.2 图形坐标的获取指令 ginput

前面所讲的图形绘制，都是先有一组表示图形的数据，然后再用适当的图形指令把这组数据表现出来。这就是计算数据的可视化过程。本节要介绍一个逆过程指令 ginput，它可以用来获取当前图形点的坐标数据。它的主要使用格式有：

[x, y] = ginput(n)	在当前坐标系统中，可选定 n 个点，并返回两个 n 维列向量。
[x, y] = ginput	在当前坐标系统中，可选任意多个点，并返回两个相应维数的列向量。

说明：

(1) 该指令只适用于二维图形。

(2) 该指令既允许用鼠标(左键或右键)选取坐标系统中的任意点，也允许用键盘上的箭头键选取。但当计算机系统中有鼠标时，该指令只能通过鼠标去选取坐标系统中的点。

(3) 在第一种使用格式中，当鼠标选满 n 点时，该指令的执行过程自动结束。在第二种调用格式中，当需要结束选点过程时，必须按【Enter】键。

(4) 当在图形窗中有多个子图时，该指令也能正常工作。但此时一定要注意：只能用鼠标在当前子图中去点；否则，ginput 指令返回的 x 、y 坐标值将可能发生混乱。

(5) 在借助图形进行分析、设计的工作(如自动控制系统的根轨迹设计法)中，该指令十分有用。

5.9 动态图形

在 MATLAB 的"上层"图形指令中的彗星轨线指令、色图变幻指令、影片动画指令，能很方便地使图形及色彩产生动态变化效果。

由于在 Notebook 和硬拷贝下，这种动态变化效果都无法表现，因此本节所有例题都不提供图形，而只给出有关指令。当读者在 MATLAB 指令窗中运作这些指令后，便可在图形窗中看到相应的动态图形。

5.9.1 彗星轨线

彗星轨线指令能动态地展示质点的运动轨迹。该指令的基本调用格式是：

comet(x, y, p)　　彗长为 p * length(y) 的二维彗星轨线。p 的缺省值为 0 1 。

comet3(x, y, z, p)　　彗长为 p * length(z) 的三维彗星轨线。p 的缺省值为0 1 。

说明：

(1) 在二维彗星轨线指令中，参数 x , p 都可以缺省。

(2) 在三维彗星轨线指令中，参数 x , y , p 都可以缺省。

【例 1】二维彗星轨线。

```
t = -pi:pi/200:pi;
comet(t,tan(sin(t))-sin(tan(t)))
```

5.9.2 色图的变幻

MATLAB 为颜色的动态变化提供了一个指令 spinmap 。它的功能是使当前图形的色图做循环变化，以产生动画效果。与前面的动态轨迹线不同，该指令不涉及图形对象特性的操作，而只限于对色图的操作。spinmap 的调用格式有以下几种·

spinmap　　使色图周期旋转约 3 s。

spinmap(t)　　使色图周期旋转为 t s。

spinmap(inf)　　使色图无限制旋转下去，用【Ctrl + C 】键中断。

spinmap(t, inc)　　分别用 t , inc(缺省值为 2) 控制色图旋转的时间和快慢。

【例 1】伪彩图的色彩变幻。

```
pcolor(peaks)
spinmap
```

5.9.3 影片动画

MATLAB 支持影片动画(movie)的制作和放映。影片动画的制作由 moviein 、getframe 指令实现。具体步骤是：

(1) 由 M=moviein(n) 创建一个具有 n 列的矩阵 M,准备用于存储 n 帧画面。这一步并非不可缺少,但是如果事先没有这样的矩阵,那么每次显示画面时都必须产生一新列,这将大大降低速度。图形的复杂程度不影响矩阵的大小,存储一帧画面的列的长度取决于图形窗口的大小,图形窗口越大占用的存储空间就越多。

(2) 用 M(·,j)=getframe 把制作影片动画用的第 j 帧画面像素以列的方式存储在矩阵 M 中。这一步主要是预先开辟内存空间,以防止占用太多的内存。

(3) 运行影片动画放映指令:movie(M,k) 。它使得在矩阵 M 中存储的画面连续播放 k 次。

【例 1】实现行波的制作和放映的命令文件。

wave1.m

```
% 制作和放映行波的命令文件
n=12;
m=moviein(n);
t=0:2*pi/n:4*pi;
x=0:pi/12:4*pi;nj=length(x);
for i=1:n
    for j=1:nj
        y(j)=sin(x(j)-t(i));
    end
    plot(x,y)
    axis([0,4*pi -1 5,1 5])
    m(:,i)=getframe;
end
movie(m,20)
```

5.10 图形的输出和打印

MATLAB 对图形的输出一向特别注重。在它的早期版本里,一方面可以用“拷屏”的方法很方便地输出图形的硬拷贝;另方面提供一个专用的 GPP 图形后处理软件,为用户提供质量较高的图形打印输出。有关早期版本的图形输出,本节不再介绍,请用户查看早期版本的用户手册。

从 4 0 版起，MATLAB 的图形输出方式发生了根本的变化：一是，可以很容易地在 Windows 平台上获得不同质量的 MATLAB 图形输出；二是，MATLAB 也提供了把图形转化为页面描述(Postscript)文件的函数指令。这种由页面描述语言表达的图形在 Ghostscript 的处理下，可给出达到专业印刷质量的图形。

不管采用上述何种图形输出方式，都有以下共同点：

(1) 在屏幕表现上存在的图形窗内的 uimenus 、uicontrols 控制键都不可能被硬拷贝。

(2) 不管在屏幕上的图形背景是黑还是白，所得硬拷贝的背景色总是白的，除非用户自己用图柄操作的底层语言进行重新设置。

(3) 图形对象上的不同彩色由单色打印机表现为不同灰度。

5.10.1 使用 Windows 应用程序打印

用图形窗菜单打印

这不仅是最简便的图形硬拷贝方式，而且所得图形中的汉字也能被正确拷贝。

(1) 在待拷贝图形(Figure)窗的【File】下拉菜单中选择【Printer Setup】，可以实现对打印机的设置(包括打印方向、纸张处理等)。

(2) 在待拷贝图形(Figure)窗的【File】下拉菜单中选择【Print】，会出现打印对话框。在这对话框里有三种打印分辨率(120dpi × 180dpi ；180dpi ；360dpi × 180dpi)可供选择。

在 MATLAB 指令窗里打印

这是在 MATLAB 指令窗里通过 Print 指令实现对 Windows 打印驱动程序和打印管理程序的控制的。具体如下：

(1) 向打印机输出图形的指令格式

这里所介绍的三个指令格式所输出的图形质量与用图形窗菜单打印的相同，不同的仅是控制方式。

print　　　　不经任何设置，直接打印当前图形窗中的对象。

print-dwin　　与上一个指令(print)作用相同。

print-v　　　通过打印对话框设置选择后，再输出当前图形窗中的对象。

(2) 引出打印机设置的指令格式

print －dsetup　　引出打印机设置对话框。

(3) 向剪贴板输出图形的指令格式

print －dmeta　　以 meta 文件格式向剪贴板输出图形。

print －dbitmap　以 bitmap 文件格式向剪贴板输出图形。

Simulink 方框图的打印

Simulink 工作时，一般会出现多个(包括模型、信号)图形窗。在这种情况下，通过相应窗口【File】下拉菜单中的【Print】选项去获得图形硬拷贝，直接方便、不容易出错。

如果想通过指令窗命令实现图形硬拷贝，则有两个问题要注意：

（1）硬拷贝模型时应采用以下指令：

print-s<ModelName>　　用（不带扩展名的）模型文件名加 s 前缀，实现模型图打印。

（2）若存在多个信号图形窗，则可以用图形窗的句柄去控制选择，指令为：

print-f<Handel>　　利用图形窗 Handel 句柄（一般是正整数 1，2，3 等）硬拷贝。

5.10.2　图形的专业印刷质量拷贝

获得专业印刷质量的图形硬拷贝的工作分两步：先把图形变成通用性的页面描述文件；后把页面描述文件转化为适用于不同打印机的格式文件。当然，这两步工作可以在不同的时间、不同的机器上进行，为图形硬拷贝的获取提供了很大方便。

GhostScript 是一个共享性软件，由它所得图形硬拷贝的质量为国际公认。GhostScript 有很强的适应性，它可适配于很多不同种类、不同档次的系统和机型。GhostScript 软件中的 use.doc 文件中详细叙述该软件在各种不同场合的使用方法。为解释简单起见，本节仅以 PC 机、Epson LQ－1600K 打印机、MS－dos 为例作具体情况介绍。当然，用户根据本节介绍，不难举一反三。

页面描述文件的生成

MATLAB 是借助 print 指令把当前图形窗中对象转化为页面描述文件的。具体格式为：

print [-ddevice] [-options] <filename>

在本节约定的环境下，对该指令后的三个选项作如下说明：

（1）<filename> 选项是必不可少的。它是将生成的页面描述文件的正名，由用户指定。而其扩展名会自动生成。假若，在此选项之前没有 [-ddevice] 选项，那么自动生成的扩展名为 .ps 。假若，在此选项之前没有 [-options] 选项，那么该指令的执行将覆盖已有的同名文件。

（2）对于 Epson LQ-1600K 打印机来说，[-ddevice] 选项常写为 -dps 。当然，-dps2 、-deps、-deps2 也是三个合法的选择。

（3）[-options] 选项常用的选择有以下三个：

-append　　在已有文件中再增加新一幅图形，而不覆盖原有图形。

-s<ModelName>　　模型文件名加 s 前缀，把 Simulink 中模型图转化为页面描述文件。

-f<handle>　　图形窗句柄加 f 前缀，把 Simulink 中的图形转化为页面描述文件。

【例 1】产生页面描述文件的一些合法完整指令示例。

print mytry1　　产生当前图形窗对象的页面描述文件 mytry1.ps 。

print -dps mytry1　　作用与上一个指令相同。

print -dps -append mytry1　　在已有的 mytry1.ps 文件上添加当前图形窗中的对象内容。

print -dps -smd1 mytry2　　产生模型名为 md1 的模型方框图的页面描述文件 mytry2.ps。

print -dps -f2 mytry3　　产生图形窗（Figure No.2）中对象的页面描述文件 mytry3.ps 。

print -deps mytry1　　产生当前图形窗对象的页面描述文件 mytry1.eps 。

说明：上述指令也能在 MATLAB Notebook 中正常运作。

环境配置

这里的环境设置是指：在PC机上，利用GhostScript 软件和 Epson LQ-1600K 获得图形硬拷贝所需的的软硬件配置。

(1) GhostScript 软件的安装

MathWorks 公司销售的 4 0 以后版 MATLAB 软件的 ghostscr 子目录上包含了 GhostScript 软件。在 MS-DOS 下使用 GhostScript 软件，用户必须作适当的软件设置。现介绍两种设置方法：

方法一，只对 autoexec bat 作如下修改：

(A) 假若 MATLAB 确实在C盘上，那么在 path 路径上加入 c：\ matlab \ ghostscr \ bin 。

(B) 再设置环境变量 set gs-lib＝c：\ matlab \ ghostscr \ ps-files；c：\ matlab \ ghostscr \ fonts 。

方法二，创建新目录 gs：

(A) 把 ghostscr \ bin 、ghostscr \ ps-files 、ghoststscr \ fonts 子目录下的所有文件拷贝到新目录下。

(B) 把需要打印的页面描述文件拷贝到新目录下。

注意：若采用方法一设置，那么不管需要打印的页面描述文件在哪里，都可以在页面描述文件所在的子目录里调用 GhostScript 软件。若是采用方法二设置，那么调用 GhostScript 软件去获得硬拷贝时，必须在新建目录 gs 下进行。

(2) 打印机的设置

国内销售的 EPSON LQ-1600K 打印机，一般都设置在中文打印状态，即打印机背后的 DIP 开关 1-6 处于 OFF 状态。为了获得正确的图形输出，必须把此开关设置成 ON(即西文打印状态)。

进入 GhostScript 环境

由于 MATLAB ghostscr GhostScript 软件子目录下提供的是 gs386 exe，因此，GhostScript 软件须在 MS-DOS 环境下运行。GhostScript 软件的启动指令的一般格式是：

gs386 -sDEVICE＝xyz -r＜xres＞x＜yres＞

说明：

(1) 输出装置开关项 -sDEVICE＝xyz

(A)该开关项可以忽略。忽略时，图形输出将送往第一优先输出装置，通常是显示器。

(B)该开关项中的 DEVICE 必须是大写字母，否则开关可能不起作用。

(C)该项中的 xyz 是输出装置名，比如 epson 、vga 等。若输出装置是打印机，那么该指令运作后，屏幕背景仍将是黑色。若输出装置是显示器，则屏幕背景将会由黑转白。

(2) 分辨率开关项 -r＜xres＞x＜yres＞

(A)该项也可以忽略。忽略时，采用不同装置的缺省分辨率。

(B)该项中的＜xres＞、＜yres＞分别是横向、纵向分辨率；而它们之间用英文字母 x 分隔。分辨率的高低直接影响硬拷贝的质量。

(3) 在这指令运行后，屏幕上将出现 GS＞ 提示符。

(4) 键入 quit 指令,则从 GhostScript 环境中退出。

在 GhostScript 环境中输出高质量的图形硬拷贝

在 GS> 提示符下,有两个常用指令:

(xyz)selectdevice　　重新指定图形输出装置 xyz 。

(PsFileName)run　　在指定的输出装置上把页面描述文件 PsFileName 转化为图形。

在这第二个指令中的 PsFileName 必须是页面描述文件全名,即要带扩展名。在此指令运行后,稍待片刻,便可得到满意的图形输出。

【例 2】在 Epson 打印机上输出高分辨率的图形的步骤。

步骤一:进入 DOS 状态,并使待处理页面描述文件(假定名为 mytry1 ps)所在目录为当前目录。

步骤二:运行下述指令进入所指定的 GhostScript 工作环境:

gs386 -sDEVICE=epson -r360x180

步骤三:运行下述指令获得文件 mytry1 ps 的相应输出:

(mytry1 ps)run

注意:步骤三中的字母 run 是必不可少的。

5.11 图形句柄的操作

到目前为止,所论及的所有绘图指令都处于 MATLAB 图形系统中的高层(high-level)界面。实际上,MATLAB 还提供了一组用于创建以及操作线、面、文字、图像等基本图形对象的底层(low-level)图形指令。这组指令可以对图形各基本对象进行更为细腻的修饰和控制,不仅可以产生更为复杂的图形,而且为动态图形的制作奠定了基础。MATLAB 的这个系统称之为句柄图形(Handle Graphics)。

5.11.1 图形对象

高层指令一般是对整个图形进行操作,但是在句柄图形中,所有的图形操作都是针对图形对象而言的。所谓图形对象是指图形系统中最基本、最底层的组元。具体地说,图形对象是指:根屏幕(Root Screen)、图形窗口(Figures)、轴(Axes)、线(Lines)、块(Patches)、面(Surfaces)、像(Images)、字(Text)、用户界面控制(User Interface Controls)和用户界面菜单(User Interface Menus)。底层指令使用户有可能对图形的一个或几个对象进行独立的操作,而不影响图形的其他部分。正是这种功能为图形操作提供了极大的灵活性。

MATLAB 的图形对象体系可用图 5 11-1 的树的结构表示。

根屏幕(Root)是整个树的根,对应于计算机屏幕,在 MATLAB 图形系统中只有一个根,其它所有的对象都是它的后代。

图形窗口(Figure)是根屏幕上的窗口,窗口的数目不限。所有图形窗都是根屏幕的子代,

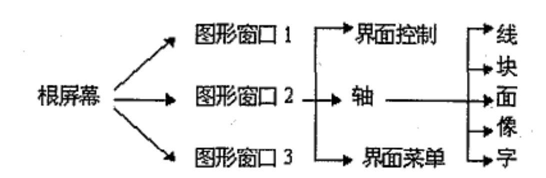


图 5 11-1 图形对象树

除根外的其他对象则是窗的后代。所有对象创建函数和高层作图指令都会建立一个图形窗口。当然，用户也可以用 figure 指令直接创建一个图形窗口。

轴(Axes)在窗口内建立一个域，并对在此域内的子代定位。轴是窗的子辈，而是线、面、块、像、字的父辈。所有对象创建函数和所有高层作图指令都能产生一个轴。轴也可以直接由 Axes 指令创建。

线(Line)是创建大多数二维图形和一部分三维图形的基本组元。线对象可以由函数 plot、plot3 、contour 和 contour3 创建。

块(Patch)是填色多边形。它也是轴的子辈，其位置决定于轴所建立的坐标系。它可以用单色或插补色进行渲染。fill 和 fill3 建立块对象。

面(Surface)是矩阵数据的三维空间表现。它由许多四边形组成，四边形的顶点又由所给数据定位。面可以单色或插补色表现，也可以仅用点间连线表现。它是轴的子辈，其位置决定于轴所建立的坐标系。pcolor 、mesh 、surf 类指令都能创建面。

像(Image)是矩阵元素直接映射到当前的色图上所得的结果。像有自己的色图。像是一个二维图形，没有观察角的调整问题。它是轴的子辈，其位置决定于轴所建立的坐标系。象由函数 image 创建。

字(Text)即为字符串。它们也是轴的子辈，位置由轴坐标系决定。

用户界面控制(Uicontrol)允许用户借助鼠标指针使选中的那个功能发挥作用。它是图形窗的子辈，独立于轴。

用户界面菜单(Uimenu)允许用户在图形窗口的上方建立菜单。它也是图形窗的子辈，独立于轴。

5.11.2 图形对象的句柄

什么是句柄

每个具体的图形对象从它产生那时起就被赋以与众不同的识别标记，这种识别标记就是该图形对象的句柄(handle)。像 contour 指令产生的多条等位线中的每一条，都有自己的句柄。

根屏幕的句柄总是零，图形窗口的句柄为某一整数，而其他对象的句柄则是浮点数。需要注意的是：由于精度原因，最好把浮点数句柄赋给变量，以备后用；千万不要试图从键盘输入从屏幕所见句柄的浮点数。

所有能创建图形对象的 MATLAB 函数都可给出所创对象的句柄。如高层指令 surfc ，既

给出"面"的句柄，又给出"线"的句柄。而底层指令 surface 只给出"面"的句柄。

对象创建函数

在 MATLAB 中，除根屏幕 Root 外，所有的对象都由与之同名的内部函数(Buıld-ın Functıons)创建。表 5 11-1 列出了这些底层函数的功能及调用方式，每一个函数都返回相应的句柄 h 。

表 5.11-1　创建对象的底层函数

函数名	功能	调用格式举例
fıgure	创建图形对象	h＝fıgure(n) n 为整数。
axes	创建轴对象	h＝axes('posıtıon',[left, bottom, wıdth, heıght]) 定义坐标盒的位置和尺寸。
lıne	画线	h＝lıne(x, y, z) 绘制向量 x、y 、z 确定的直线。如果不指定 z ，则在 x-y 平面上画线。
patch	填充多边形	h＝patch(x, y, z, c) x 、y 、z 定义多边形，c 指定填充颜色。
surface	创建三维曲面	h＝surface(x, y, z, c) x 、y 、z 定义三维曲面，c 是色彩矩阵。
ımage	显示图像	h＝ımage(x) x 为图像矩阵。
text	标注文字	h＝text(x, y, 'strıng') x 、y 指定字符串 strıng 的标注位置。
uıcontrol	用户界面控制	h＝uıcontrol('property', value) property/value 指定界面的控制类型。
uımenu	用户界面菜单	h＝uımenu('property', value), property/value 指定图形窗口上方的菜单形式。

每个底层函数只能创建九个图形对象中的一个，并将它们置于适当的父辈对象之中。例如，lıne 指令的运作将在当前轴上利用缺省特性数据画"线"。假若，此指令运作前，"轴"、"窗"不存在，则 MATLAB 会自动创建它们。假若，此指令运作前，"轴"、"窗"已经存在，那么这"线"将被画在已有的"轴上"，且不影响该轴上已有的其他对象(注意，这与高层作图指令不同!)。这个特点非常重要，特别是当图形仅有某一部分需要改变时。

表 5 11-1 所列的内部函数是 MATLAB 的作图核心程序。MATLAB 的高层作图函数都建立在它们的基础之上。下面通过解剖高层作图指令 surf m 文件，说明它是如何依赖底层作图函数 surface 建立起来的。

【例 1】高层作图函数 surf m 文件与底层作图指令 surface 的关系。

surf.m

```
function h = surf(x, y, z, c)
cax = newplot;                    % 调用 newplot m ,建立新图形窗口
if nargin = = 0
      error('Not enough input arguments ')
elseif nargin = = 1               % 一个输入宗量情况
     if min( size( x ) ) = = 1
        error('Input argument must be a matrix not a vector or a scalar')
```

```
    else
      hh = surface(x);           % 利用单参数格式创建“面”
    end
elseif nargin == 2               % 两个输入宗量情况
    hh = surface(x,y);           % 利用双参数格式创建“面”
elseif nargin == 3               % 三个输入宗量情况
    hh = surface(x,y,z);         % 利用三参数格式创建“面”
elseif nargin == 4               % 四个输入宗量情况
    hh = surface(x,y,z,c);       % 利用四参数格式创建“面”
else
    error('Too many input arguments ')
end
next = lower(get(cax,'NextPlot'));       % 把大写字母变为小写
if ishold
    view(3);                     % 非 hold 状态下,用默认视角观看三维图形
end
if nargout == 1                  % 有一个输出参数时,将 surface 的句柄赋给变量 h
    h = hh;
end
```

对象句柄的获得与删除

在本小节的例 1 中,已经看到:通过指令的赋值调用格式 h = surface ,就可在对象创建的同时,获得创建对象 surface 的句柄值 h 。这是获得对象句柄的一种方法。

MATLAB 还提供两个专门用于获取对象句柄的函数:

gcf　　返回当前图形窗口的句柄。

gca　　返回当前轴的句柄。

它们既可以赋值给某一个变量以供后用,也可直接作为其他函数的输入宗量使用。例如指令 delete(gca) 将删除当前的轴以及它的所有子代。(函数 delete 可以删除任何对象,也能够删除文件,这里不再介绍。)

5.11.3 对象品性

什么是对象品性

所有对象都有决定其如何表现的品性。对象品性可分为两类:一类是“共性”,包括它的父辈和子代(Parent、Children)、类型(Type)、是否可视(Visible)、剪辑(Clipping)、中断允许(Interruptible)等;另一类是“特性”,如“轴”的刻度、定义“面”的数据等。

当对象被创建时,该对象的品性就被一组缺省值初始化。

用户既可以(用 get 指令)查询任何品性值,也可以(用 set 指令)设置大多数的品性值,且

设置值仅适用于本对象，而对其他同类对象的品性值没有影响。

表 5 11-2 到表 5.11-10 列出了各种对象的常用品性名称和可赋值。在这些列表的品性值栏中，黑体字表示 MatheWorks 公司的“厂家设置值(Factory Setting Values)”。色彩品性值栏中常用到色彩性质 ColorSpec 。ColorSpec 不是指令，而是指色彩选择的如下三种方法：

(1) RGB 三元数组法，如 [1 , 1 , 1] 代表“白”、[0 , 0 , 0] 代表“黑”等，详见第 5 7 1 节。

(2) 色彩全名法，如 yellow , magenta , cyan , red , green , blue , white , black 。

(3) 色彩简名法，如 y , m , c , r , g , b , w , k 。

表 5.11-2 “根屏幕”的品性

品性名 PropertyName	含 义	品 性 值 PropertyValue
CurrentFigure	当前图形	“图形窗口”句柄，正整数
Diary	日记是否打开	on , **off**
DiaryFile	日记文件名	字符串
Echo	回显开关	on , **off**
Format	数据输出格式	**short** , long , short e , long e ,
FormatSpacing	输出空格状态	compact , **loose**
Clipping	图形剪辑方式	**on** , off
Interruptible	是否允许用户中断	yes , **no**
Visible	是否可视	**on** , off

表 5.11-3 “图形窗口”的品性

品性名 PropertyName	含 义	品 性 值 PropertyValue
Color	窗口颜色	ColorSpec, RGB [**0 0 0**] , **即(k)**
Colormap	色图	RGB 值的(64×3)矩阵
CurrentAxes	当前轴	“轴”的句柄
CurrentObject	当前对象	光标所指对象的句柄
InvertHardcopy	图形硬拷贝时反色否	**on** , off
MenuBar	是否列菜单	**on** , off
NextPlot	下一个图形的绘制方式	new , **add** , replace
NumberTitle	Windows 窗口是否用句柄做图名	**on** , off
PaperOrientation	横打或竖打	**portrait** , landscape
PaperPosition	打印纸的位置	**[0 25 2 5 8 6]**
PaperSize	打印纸大小	**[8 5 11]**
PaperUnits	纸张大小的计量单位	normalized , **inches** , centimeters ,
PaperType	纸的类型	**usletter** , uslegal , a4letter
Pointer	鼠标指针的样式	**arrow** , crosshair , watch , cross ,
ScreenSize	图形窗的位置和大小	四元向量 [left , bottom , width , heidth] **[1 1 640 480]**
Resize	图形窗大小可改否	**on** , off
ShareColors	公用颜色	**yes** , no
Units	绘图计量单位	**pixels** , normal , inches ,

表 5.11-4 “轴”的品性

品性名 PropertyName	含 义	品 性 值 **PropertyValue**
AspectRatio	视图比例	[纵横轴比，刻度比]，[NaN NaN]
Box	坐标轴是否成封闭状	on , **off**
CLim	色轴范围	[cmin , cmax] , [**0 1**]
CLimMode	定义色轴范围的模式	**auto** , manul
Color	坐标轴定义区的颜色	**none** , ColorSpec
ColorOrder	画“线”的用色次序	RGB 的（m×3）矩阵，缺省用色次序为 **yellow** , **magenta** , **cyan** , **red** , **green** , **blue**
DrawMode	绘图快慢方式	**normal** , fast
FontName	字体名字	**Helvetica**
FontSize	字体大小	[**12**]
GridLineStyle	分格线的线型	- , -- , · , -
LineWidth	线宽	**0 5**
NextPlot	如何画下一幅图	new , add , **replace**
Position	轴的位置	[left , bottom , width , height]
TickLength	轴上刻度记号的长度	[2 维用，3 维用]，[**0 01 0 025**]
TickDir	刻度朝向(向内、外)	**in** , out
Units	单位	pixel , **normalized** , inches ,
View	图形视角	[方位角，俯视角]
XGrid	是否画 x 方向分格线	on , **off**
XLabel	x 轴的标注	“字”句柄
XScale	x 轴的刻度方式	**linear** , log
Y··	y 轴品性控制	与 x 轴相同
Z	z 轴品性控制	与 x 轴相同

表 5.11-5 “线”的品性

品性名 PropertyName	含 义	品 性 值 **PropertyValue**
Color	线的色彩	ColorSpec, RGB [**0 0 0**]，即(**k**)
EraseMode	擦除方式	**normal** , none , xor , backgound
LineStyle	线型	**-** , -- , : , - o , + , , * , x
LineWidth	线宽	标量数值,(**0 5**)
MarkerSize	标志大小	标量数值,(**6**)

表 5.11-6 “面”的品性

品性名 PropertyName	含 义	品 性 值 **PropertyValue**
CData	决定 ZData 色彩	数值矩阵
EdgeColor	网格线的颜色	ColorSpec , none , **flat** , interp (**k**)
EraseMode	擦除方式	**normal** , none , xor , backgound

表 5.11-6(续)

品性名 **PropertyName**	含　义	品 性 值 **PropertyValue**
FaceColor	网格表面颜色	ColorSpec , none , **flat** , interp
LineStyle	线型	- , -- , 　, - 　o , + , 　, * , x
LineWidth	线宽	标量数值, **(0 5)**
MarkerSize	标志大小	标量数值, **(6)**
MeshStyle	经纬网格线模式	**both** , row , column

表 5.11-7　"块"的品性

品性名 **PropertyName**	含　义	品 性 值 **PropertyValue**
CData	决定"块"边色彩	数值向量
EdgeColor	网格线的颜色	ColorSpec , none , **flat** , interp **(k)**
EraseMode	擦除方式	**normal** , none , xor , backgound
FaceColor	网格表面颜色	**ColorSpec** , none , **flat** , interp

表 5.11-8　"字"的品性

品性名 **PropertyName**	含　义	品 性 值 **PropertyValue**
Color	"字"的颜色	ColorSpec **(w)**
FontName	字体名称	**Helvetica**
FontSize	字体大小	**[12]**
FontUnderline	字符串加下划线否	on , **off**
HorizontalAlignment	字与设点的水平关系	**left** , center , right
Rotation	旋转角度	−270 , −180 , −90 , **0** , 90 , 180 , 270
VerticalAlignment	字与设点的垂直关系	top , cap , **middle** , baseline , bottom

表 5.11-9　"用户界面控制 Uicontrol"的品性

品性名 **PropertyName**	含　义	品 性 值 **PropertyValue**
BackGroundColor	控制钮的背景颜色	ColorSpec
CallBack	控制钮功能设定	字符串
ForeGroundColor	控制钮名字符颜色	ColorSpec
HorizontalAlignment	控制钮名的位置	left , **center** , right
Position	控制钮在图形窗位置	四元向量 [left , bottom , width , height]
String	给控制钮命名	用"|"分隔的字符串组
Style	控制钮类型	pushbutton , radiobutton , checkbox , slider , edit , popupmenu
Units	控制钮大小计量单位	pixel , normalized , inches ,

表 5.11-10 "用户界面菜单 Uimenu"的品性

品性名 PropertyName	含 义	品 性 值 PropertyValue
Accelerator	菜单快捷键定义	PC 机定义方法.Ctrl + 字母
CallBack	菜单调用功能设定	字符串
Enable	现场允许激活变色	**on** , off
Label	菜单项命名	字符串
Position	菜单条位置	标量
Separator	菜单项划线模式	on , **off**

对象品性的设置和查询

MATLAB 提供两种设置对象品性的方法:一种是在对象创建时设置其品性;另一种是通过函数 set 重新设置已有对象的品性。

【例 1】创建图 5 11-2 所示二维图形时对象品性的设置指令示范。

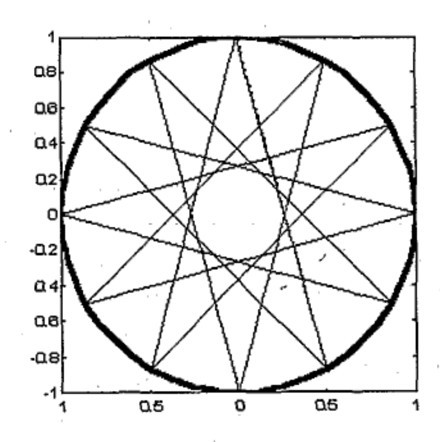


图 5 11-2 创建时设置对象品性的二维图形

```
fig=figure(1)                  % 创建 1 号图形窗,并将该窗句柄值存放于 fig
ax=axes('Box','on','AspectRatio',[1,NaN])
                               % 创建封闭式的、高宽比为 1 的坐标轴,ax 存轴的句柄值
x=0:pi/24:2*pi;
lin1=line(sin(x),cos(x),'LineWidth',5,'LineStyle','-','Color','r')
                               % 用 5 号粗实红线画圆。此线句柄存放于 lin1
lin2=line(sin(20*x),cos(20*x),'LineStyle',':','Color','b')
                               % 用(缺省)0 5 号细虚蓝线画弦。此线句柄存放于 lin2

fig =
1
ax =
```

```
   62.0007
lin1 =
    63.0007
lin2 =
    64.0007
```

【例 2】创建三维网线图时设置对象品性(图 5.11-3)。

```
fig=figure;
ax=axes('Box','on','View',[-37.5,30],'Nextplot','replace');
[x,y,z]=peaks(25);
surfh=surface (x,y,z,'Facecolor','w',..
              'LineWidth',2,'Edgecolor','k');
```

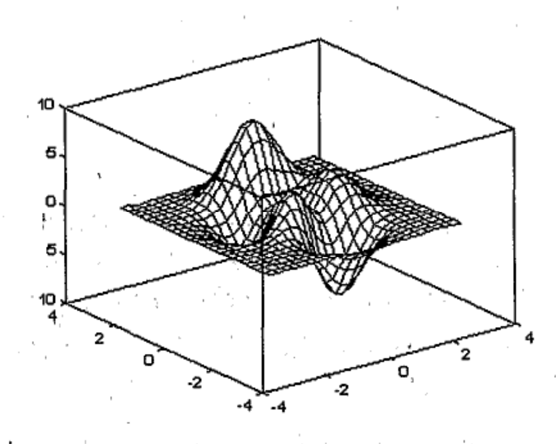


图 5 11-3 创建时设置对象品性的三维图形

说明：由于在设置中“Facecolor”设为“w”，这实际上是高层指令 mesh 所画的网线图。此图的网线比较粗。

查询函数 get

get(h) 查询 h 对象所有品性的当前值。
get(h,'PropertyName') 查询 h 对象(由 PropertyName)所指定品性的当前值。

【例 3】查询上例所创建“面”的品性。

```
get(surfh)
CData = [ (25 by 25) ]
EdgeColor = [0 0 0]
EraseMode = normal
FaceColor = [1 1 1]
LineStyle = -
```

```
LineWidth = [2]
MarkerSize = [6]
MeshStyle = both
XData = [ (25 by 25) ]
YData = [ (25 by 25) ]
ZData = [ (25 by 25) ]

ButtonDownFcn =
Children = []
Clipping = on
Interruptible = no
Parent = [58.0002]
Type = surface
UserData = []
Visible = on
```

设置函数 set

set(h) 显示 h 对象所有可设置的品性及取值。
set(h,'PropertyName') 显示 h 对象(PropertyName)指定品性的可取值。
set(h,'PropertyName',PropertyValue,...) 设置 h 对象(PropertyName)指定品性的值。

下面用两个示例来说明 set 指令的用法。从中也可体会到，MATLAB 的高层指令是如何借助底层函数建立起来的。

【例 4】通过重新设置图 5.11-3 的"面"品性获得着色表面图(图 5.11-4)。

```
set(surfh,'Facecolor','flat')
```

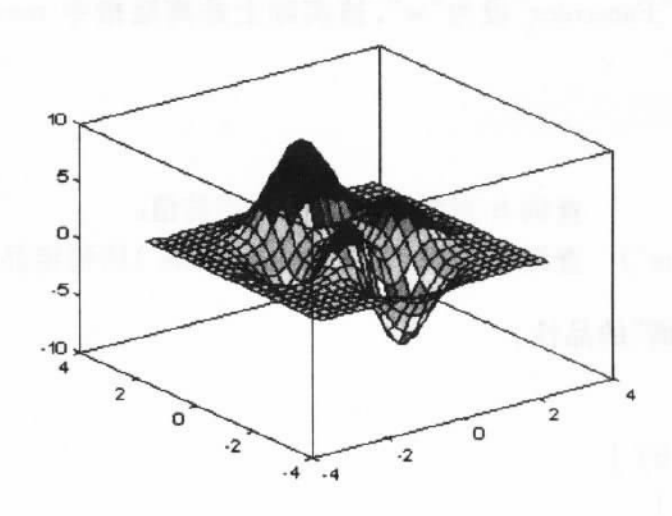
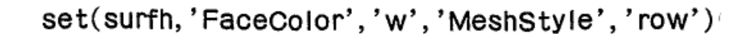


图 5.11-4 通过重新设置图 5.11-3 的"面"品性获得着色表面图

【例 5】通过重新设置图 5.11-3 的"面"品性获得瀑布线图(图 5.11-5)。



```
set(surfh,'FaceColor','w','MeshStyle','row')
```

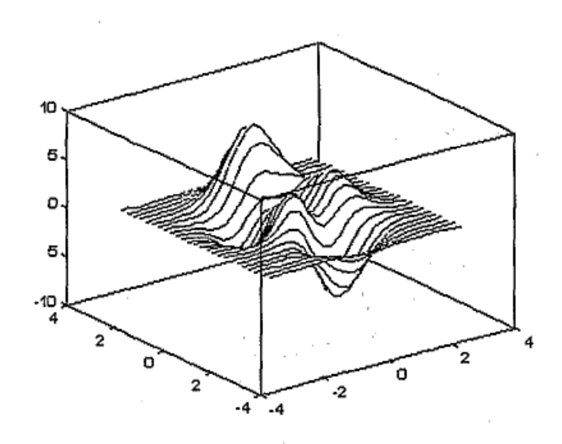


图5 11-5　通过重新设置图5 11-3的“面”品性获得瀑布线图

对象品性的缺省值

在MATLAB中对象品性有两个层次的缺省值设置：一个是由MatheWorks公司所设置的对象品性缺省值，称为“厂家设置值(Factory Settıng Values)”，这种设置值不可能被用户清除；另一种是用户根据自己需要设置的对象品性缺省值。假如不修改MATLAB的启动文件startup m，那么这种用户定义的缺省值只在当前MATLAB环境中有效。

一个对象品性对缺省值的搜寻是从当前对象开始的，假如找不到与该品性适配的缺省值，那么将沿着图形对象树向上搜索，直到发现一个适配的用户定义缺省值或厂家设置缺省值为止。

用户缺省值是通过set函数实现的。其一般格式如下：

set(h,'DefaultObjectTypeObjectProperty',PropertyValue)

说明：

(1) h是图形对象的句柄，它将决定用户缺省值的使用范围。

(2) 描述符DefaultObjectTypeObjectProperty表示，设置对象缺省值的语法结构，即“Default + 在h对象中需设置缺省值的对象名 + 那对象的品性名”。前导符“Default”是必须有的。

(3) PropertyValue指将设的用户缺省值。

关于缺省值的设置、使用和清除，用以下示例说明。

【例6】

```
set(0,'DefaultFigureColor','w')
set(gcf,'DefaultLineColor','k')
```

说明：

(1) 第一条指令的作用是：把“根”上所开的“图形窗口”的颜色缺省值设为“白色”。这将导致以后所有的图形窗口都是白色背景。

（2）第二条指令的作用是：把当前“图”上的“线”的颜色缺省值设为“黑色”。

【例 7】缺省值的设置与使用。，

```
h=surf(peaks)
set(0,'DefaultSurfaceEdgeColor','g')
set(h,'EdgeColor','default')
```

说明：

（1）第二条指令在“根”上，将“面”的网线颜色的缺省值设置为“绿色”。

（2）第三条指令驱使第一条指令画表面网线时使用缺省色。经搜索，网线颜色缺省值就是前句所定义的“绿色”。因此，这三条指令运行后所画的曲面采用绿网线。

【例 8】使用“首遇”缺省值。

```
h=surf(peaks)
set(gca,'DefaultSurfaceEdgeColor','r')
set(0,'DefaultSurfaceEdgeColor','g')
set(h,'EdgeColor','default')
```

说明：

（1）该例是在例 7 基础上加进了一条 set(gca,'DefaultSurfaceEdgeColor','r')指令。这是在当前“轴”上，设置“面”的网线缺省值为“红”。

（2）这四条指令的运行结果与例 7 不同，所画曲面的网线是红的。这是因为当最后一句指令去寻找网线缺省值时，先在 h“面”上搜索；没找到，再沿树往上到“轴”搜索；在“轴”上找到了对网线定义的缺省值“红色”；于是，停止搜索，并用这“首遇”缺省值（“红色”）去画曲面网线。本例情况下，第三句指令的设置对所画图形不起作用。

【例 9】如何使用“厂家设置值”。

```
h=surf(peaks)
set(gca,'DefaultSurfaceEdgeColor','r')
set(0,'DefaultSurfaceEdgeColor','g')
set(h,'EdgeColor','factory')
```

说明：不管如何设置缺省值，由于最后一句指令指定当前曲面用“厂家设置值”画网线，所以这四句指令执行后，将得到黑网线曲面。

【例 10】利用“remove”清除用户所设缺省值。

```
set(0,'DefaultSurfaceEdgeColor','remove')
```

【例 11】在不同“层面”上设置缺省值的影响（图 5.11-6）。

```
figure('Position',[360,250,950,400])
set(gcf,'DefaultAxesBox','on')              % “图形窗”上定义“轴”缺省值为封闭
subplot(1,2,1)                              % 开辟“图形窗”的左半区
```

```
set(gca,'DefaultLineLineStyle',':')        % 在"轴"上定义"线"缺省值为虚线
h1=plot(sin(0:pi/20:2*pi))
set(h1,'LineStyle','default')
hold on
h2=plot(cos(0:pi/20:2*pi))
set(h2,'LineStyle','default')
text('Position',[20,0 4],'String','sine')
text('Position',[15,-0.3],'String','cosine',..
    'HorizontalAlignment','right')         % 在当前轴上水平写"字"

subplot(1,2,2)
set(gca,'DefaultTextRotation',90)          % 在"轴"上定义"字"旋转 90 度
plot(sin(0:pi/20:2*pi))
hold on
plot(cos(0:pi/20:2*pi))
text('Position',[20,0 4],'String','sine')
text('Position',[15,-0 3],'String','cosine',.
    'HorizontalAlignment','right')
h1 =
  57.0001
h2 =
  58.0001
```

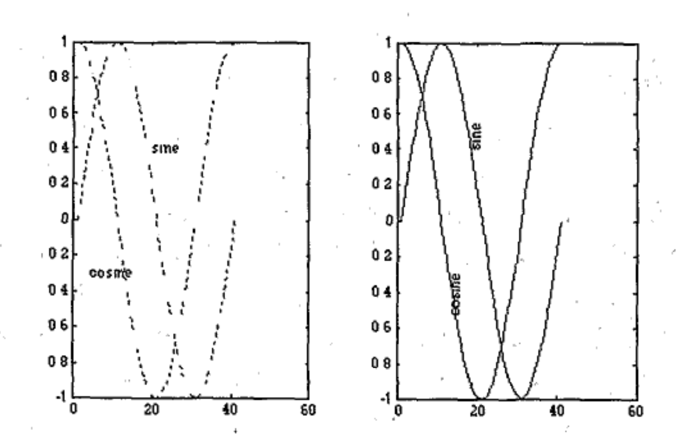


图 5 11-6 不同"层面"缺省值的影响

说明：由于"轴"缺省值(封闭轴)定义在"图形窗"上，因此用户缺省值可被该窗中的任何轴所引用。而那些在"轴"上定义的"线"、"字"缺省值只可被所在"轴"上的对象引用。上述两个子图间的不同清楚地说明了这一点。

操纵对象品性的高层指令

前面介绍的 get 、set 是对各种对象的各种品性进行查询和设置的通用指令。除此以外，MATLAB 还提供了一组易于操纵的高层指令(见表 5 11-11)。这种高层指令都是针对常用对象的一个或几个品性设计的。它们简明、方便,更接近用户习惯。

表 5.11-11 操纵对象品性的高层指令

高层指令	操纵对象名	可设置与提取的对象品性
axis	axes	Xlim , Ylim , Zlim XLimMode , YLimMode , ZLimMode View , Ydir , Position , Visibility AspectRatio
caxis	axes	Clim , CLimMode
cla	axes	deletes Children
clf	figure	deletes Children
colormap	figure	ColorMap
gca	figure	CurrentAxis
gcf	root	CurrentFigure
grid	axes	Xgrid , Ygrid , Zgrid
hold	axes	NextPlot
orient	figure	PaperOrientation , PaperPosition
reset	axes, figure	all except Position
subplot	figure axes	NextPlot , CurrentAxis Position
view	axes	View , XForm

5.11.4 实时动画的制作

图形从静到动是质的飞跃。它要求用户有更全面的知识和灵感,同时也给用户提供科学和美相互协调的全新想象空间,从而激发出前所未有的创造力。

动画的挑战意义还在于,制作动画时所面对的软硬件资源限制。从硬件上讲,图形需占较多的计算机资源。对 VGA 以上的彩显来说,大量的图形操作会明显降低计算机的运行速度,因此一个高性能的计算机在动画制作中是必不可少的。从软件方面讲,动画制作通常有两种方法:(1)预先将图形制作好,并放到图形缓冲区内,然后一帧一帧地播放。这种方法计算量大、占用内存多。要达到人眼不感到停滞的连贯程度,播放速度至少 24 帧/s。这对计算机硬件是不低的要求。目前的多媒体技术就用这种方法,前面讲的 MATLAB 中的影片动画制作也用这种方法。(2)保持整个背景图案不变,只更新运动部分的图案,以便加快整幅图像的实时生成速度,这称为实时动画。很多画面精美的计算机游戏都采用这种技术。这正是本节要介绍的内容。

改变某几个图形对象而不破坏其它部分图形的具体办法是:(1)计算活动对象的新位置,并在新位置上将它显示出来;(2)擦除原位置上原有对象,刷新屏幕;(3)重复步骤 1 和步骤2。

如何显示新对象，擦除旧对象，而又不破坏背景图案，这对一般计算机编程语言来说不是件轻松事。然而，这在 MATLAB 图形系统中却轻而易举，只需在创建图形对象时指定它的擦除方式(EraseMode)便可。

MATLAB 为 EraseMode 品性设置了四个选项：

(1)normal 方式

使用该选项后，重画整个显示区。这种模式产生的图形最准确、但较慢。

(2)background 方式

将旧对象的颜色变为背景颜色，从而达到擦除的目的。这种模式将损坏被擦对象下面的对象。

(3)xor 异或方式

对象的绘制和擦除由该对象颜色与屏幕颜色的异或而定。只画与屏幕色不一致的新对象点；只擦与屏幕色不一致的原对象点。该方式不损害被擦对象下面的其他图像。

(4)none 方式

不作任何擦除。

当新对象品性设置后，应刷新屏幕，使新对象显示出来。MATLAB 刷新屏幕的指令是 drawnow 。drawnow 指令迫使 MATLAB 暂停目前的任务序列而去刷新屏幕；若没有 drawnow 指令，MATLAB 要等任务序列执行完后才会去刷新屏幕。

下面是两个实时动画制作实例。

【例 1】M 文件 dsine.m 可显示小球沿正弦曲线运动(图 5 11-7)。

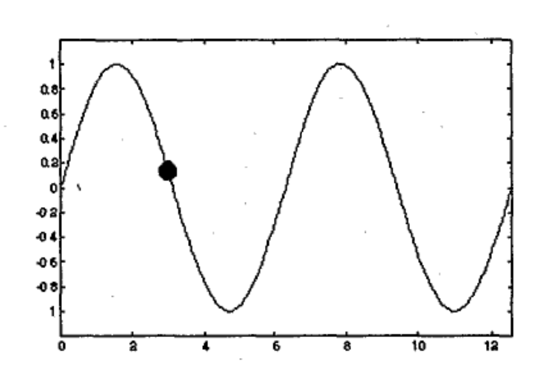


图 5 11-7　小球沿正弦曲线运动中的一个镜头

dsine.m

```
t=0:pi/48:4*pi;
y=sin(t);
plot(t, y, 'b')            % 绘制正弦曲线
% 利用“线”对象创建小球
% 定义“线”色、“线”型(点)、点的大小(50)、擦除方式(xor)
% 小球对象的句柄值保留在变量 h 中。
n=length(t);
```

```
h=line('color',[0 0 5 0 5],'linestyle',' ', ..
    'markersize',50, 'erasemode','xor');
% 使小球运动
i=1;
while 1                                    % 无穷循环
    set(h,'xdata',t(i),'ydata',y(i));      % 小球位置
    drawnow;                               % 刷新屏幕
    i=i+1;
    if i>n,
        i=1;
    end
end
```

说明：

(1) 小球的移动速度决定于 t 的增量、计算机运行窗口的多少以及计算机的运算速度。

(2) 程序无穷循环，可用 Ctrl+Break 中断。

【例 2】如图 5 11-8 所示单摆的动态演示程序的编写。

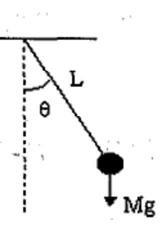


图 5 11-8

小球质量为 M，线长为 L，不考虑空气阻力的影响。当 θ 较小时，认为小球作理想简谐运动。其运动方程为：

$$\theta + \frac{g}{L\theta} = 0$$

其解为：

$$\theta = \theta_0 \cos\sqrt{\frac{g}{L}}t$$

其中，$\theta_0 = \theta|_{t=0}$。选择参数值 $g=9\ 8$，$\theta_0 = \frac{\pi}{6}$，$L=1$。

程序编制如下：

pend.m

```
% 挂摆横梁
plot([-0 2;0 2],[0;0],'color','y','linestyle','-',
    'linewidth',10);
% 画初始位置的单摆
g=9 8;
l=1;
theta0=pi/6;x0=l*sin(theta0);
y0=-l*cos(theta0);
axis([-0 75,0 75,-1.25,0]);
axis('off');            % 不显示坐标轴
% 创建摆锤
```

```
head=line(x0,y0,'color','r','linestyle',' ',
        'erasemode','xor','markersize',40);
% 创建摆杆
body=line([0;x0],[0;y0],'color','b','linestyle','-',
        'erasemode','xor');
% 摆的运动
t=0;
dt=0 01;
while 1
      t=t+dt;
      theta=theta0*cos(sqrt(g/l)*t);
      x=l*sin(theta);
      y=-l*cos(theta);
      set(head,'xdata',x,'ydata',y);
      set(body,'xdata',[0;x],'ydata',[0;y]);
      drawnow;
end
```

说明.本程序可以在 MATLAB 中运行,并在图形窗中看到单摆将无休止地运动。可用 Ctrl+Break 中断程序。

第六章　MATLAB Notebook

Mathworks 公司开发的 MATLAB Notebook 成功地把 Microsoft Word 6 0 与 MATLAB 集成为一个整体，为文字处理、科学计算、工程设计营造了一个完美统一的工作环境。它不仅拥有 MS-Word 6 0 的全部文字处理功能，而且具备 MATLAB 无与伦比的数学解算能力和灵活自如的计算结果可视化能力。它即可以看作能解决各种计算问题的字处理软件，也可以看作具备完善文字编辑功能的科技应用软件。

假如仅从字处理能力上讲，Microsoft Word 与 Word Processing、Word Perfect、LaTex 等优秀软件相比并不显得优越。假如仅从数学解算能力及应用方便上讲，即便是 Mathematica、Maple 也难以与 MATLAB 相比。至于从整体上讲，那就几乎没有一个软件可与 MATLAB Notebook 相提并论。拿流行已久的 Mathcad 来说，其字处理能力不如 Word，其计算结果可视化能力远在 MATLAB 之 下，其数学计算能力更无法与 MATLAB 相比。

MATLAB Notebook 是“活”的笔记本。在该笔记本中的计算指令可随时修改、即时解算并图示。对于撰写科技报告、论文、专著的科技工作者，对于讲授、编写理工学科教材讲义的教师，对于学习演算理工学科习题的广大学生，这“活”性是最宝贵的。

在本书前言里已经说过，本书没有手写稿，全部软件文稿是用 MATLAB 的中文 Notebook 完成的。假如拿本书的软件文稿，即本书的“电子表现形式”(Electronic Book)，与读者手中这本印刷出来的书相比，那么软件文稿有两大特色：一是软件文稿中的图形都是彩色的；二是软件文稿中的“指令细胞”都可被激活，给用户提供更好的实践机会。

6.1　入　门

MATLAB Notebook 允许用户把前台 Word 文档内创建的命令送到后台的 MATLAB 中运算，然后将运算结果送回到前台的 Word，并插入到文档中。运算结果既可以是数据又可以是图形。

MATLAB Notebook 文件被称之为 M-book 。这是因为 MATLAB Notebook 文档在 Word 中使用的模板是 M-book dot。一个 M-book 被打开时，MATLAB 随之被启动。从这个意义上说，M-book 实际上是一个交互式 MATLAB 过程的记录。这个记录包含了文字、输入输出数据和图形，并可借助 MS-Word 对之进行完善的处理。

现在请读者跟着本节的叙述来体会一下形成文稿时“活”的含义。假设读者与我们一起坐在计算机前，写以下两个算例的软件文稿。

【例 1】“活”的指令和运算结果演示。

(1) 在下一行里，从键盘敲入 MATLAB 的 12 阶魔方阵指令。

(注意：下一行的所有字符必须在英文状态下输入。)

```
x=magic(12)
```

(2) 指令字符输完后，按【Ctrl+Enter】。读者马上可以看到，指令字符颜色由黑变绿；字体也相应变化。在这同时，屏幕上出现以下(蓝色)运算结果：

```
x =
   144     2     3   141   140     6     7   137   136    10    11   133
    13   131   130    16    17   127   126    20    21   123   122    24
    25   119   118    28    29   115   114    32    33   111   110    36
   108    38    39   105   104    42    43   101   100    46    47    97
    96    50    51    93    92    54    55    89    88    58    59    85
    61    83    82    64    65    79    78    68    69    75    74    72
    73    71    70    76    77    67    66    80    81    63    62    84
    60    86    87    57    56    90    91    53    52    94    95    49
    48    98    99    45    44   102   103    41    40   106   107    37
   109    35    34   112   113    31    30   116   117    27    26   120
   121    23    22   124   125    19    18   128   129    15    14   132
    12   134   135     9     8   138   139     5     4   142   143     1
```

【例2】"活"指令和它创建的三维图形(图6 1-1)。

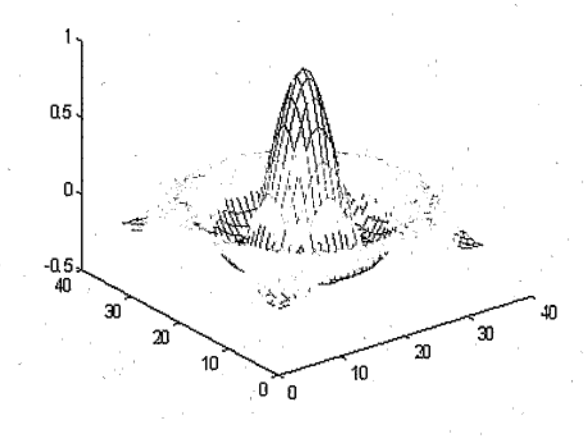


图6 1-1 在Notebook中的图

在英文状态下，先输入一组命令；然后用鼠标把它们选亮；在【Notebook】下拉菜单中选择【Define Input Cell】；再按【Ctrl+Enter】组合键，图6 1-1所示图形便产生在该组指令下方。

```
x=-8:0.5:8;
y=x';
X=ones(size(y))*x;
Y=y*ones(size(x));
```

```
R=sqrt(X.^2+Y.^2)+eps;
Z=sin(R)./R;
mesh(Z)
```

说明：该图形（在屏幕上是彩色的！）是直接生成的，而不是通过剪贴板间接获得。它的大小和位置可据需要设定。

从上述两个算例的形成过程，作以下一般性归纳：

(1) Notebook 中的 MATLAB 指令和其运算结果都是"活"的。这不仅在生成过程中如此，而且在生成后，仍然是"活"的。拿上述算例来说，在它们生成后，假如再改变指令，运算结果和图形都会作相应的改变。

(2) Notebook 创建的文字、数据及图形完全符合 Word 的规定格式。因此，可以利用 Word 的强大功能对它们进行编辑和排版。

(3) 创建后的 Notebook 是一个标准的 Word 文档，因此可以直接从 Word 中打印输出。

由于 Notebook 建立在 MATLAB 运算、绘图能力和 Word 字处理能力基础之上，用好 Notebook 的关键取决于掌握 MATLAB 和 Word 6.0 的程度。因此，下节简要介绍中文 Word 6.0 的使用。

6.2 中文 Word 6.0 简介

6.2.1 文档的创建与编辑

每当启动 Word 后，便会看到如图 6.2-1 所示的 Word 工作区窗口。该工作区包括：标题

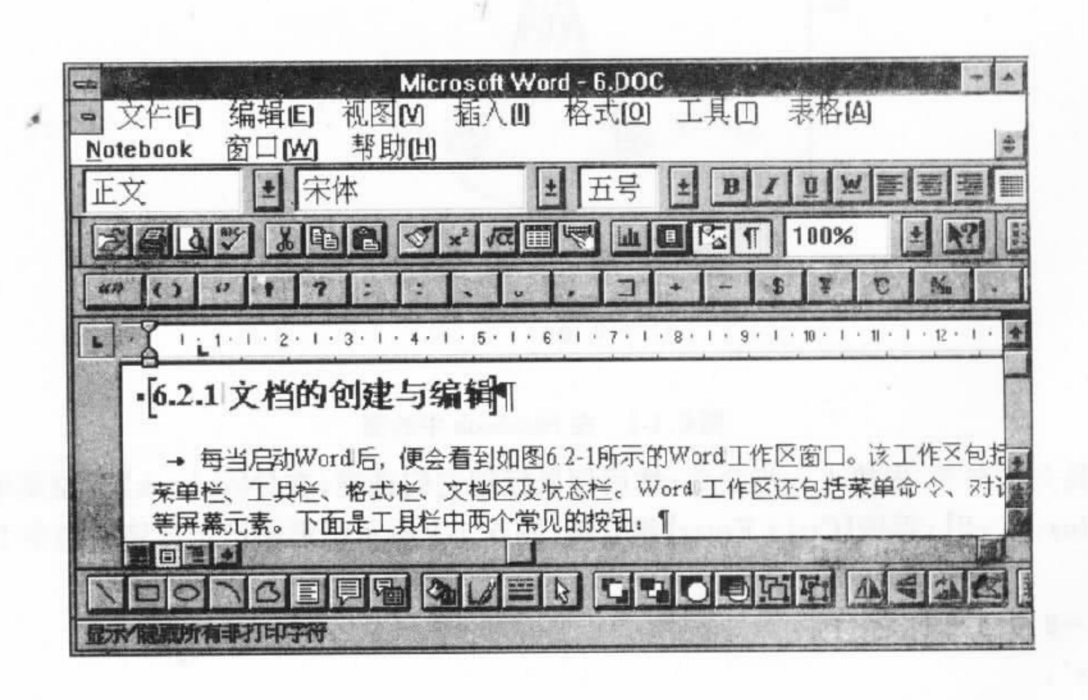


图 6.2-1 Word 工作区窗口

栏、菜单栏、工具栏、格式栏、文档区及状态栏。Word 工作区还包括菜单命令、对话框和窗口等屏幕元素。

对图 6 2-1 窗口作以下说明：

(1) 该窗口是在 MATLAB Notebook 工作状态下的布置，即调用的是 M-book dot 模板。它与普通 Normal dot 模板的外观区别是，在菜单栏中多了一个【Notebook】选项。

(2) 该窗口标题栏中“6 DOC”是本章编稿时的文件名。格式栏中“正文”、“宋体”、“五号”及最右边的变色按键表示：当时光标所在段是正文内容、采用五号宋体字、段落采用两端对齐格式。

(3) 在工具栏中，变色的非打印字符“显示/隐藏”键¶正处在显示标记状态，这在窗口最下方的状态行中有相应的说明。此时，在文档区中可看见“制表 Tab 符”、“字间空格符”和“段落符”。

(4) 文档区在标题两侧所看见的黑点和方括号是标题定义符和目录编制符。该窗口最下方状态栏中的第二个变色按钮表明，当时的视图为“页面”状态。

若把鼠标指向工具栏、格式栏的某个按钮和选项，在鼠标下方及 Word 工作区最下方的状态栏中有对该按钮的功能作简单说明。

若要详细了解某一屏幕元素，请单击工具栏上的N?帮助按钮。当鼠标指针变为箭头加问号时，指向那要了解的屏幕元素并单击之，就出现帮助信息。

每次进入 Word 时，Word 就自动创建一个新文档，并暂时命名为“文件 1”(对中文 Word 而言)。通常情况下，这个新文档默认使用模板文件 Normal dot，默认页面大小为标准 A4 纸，即宽 21 cm，高 29 7 cm。在 MATLAB Notebook 情况下，每个新文档使用的模板文件是 M-book dot。图 6 2-1 就是 Word 在 M-book dot 模板下的窗口布置。

因为 Word 总是自动创建新文档，所以一旦进入 Word，就可以立即开始输入要建立的文档内容。

输入文本

每当 Word 启动后，Word 就显示出一空白文档。称为插入点(或称光标)的闪烁竖条指示键入内容的位置。与使用打字机不同，当输入内容在一行中容纳不下时，用户无须也不必在右边界处按【Enter】键换行，而尽管继续输入，Word 会自动换行。这个特性称为“字回绕”。只有当一个段落结束时，才需按下【Enter】键。今后不管页面大小如何变化，经这样操作所获得的段落将能始终保持用户所需的格式。

假如需要，可以通过键盘操作删除插入点左右两边的字符。

当一帧屏幕显示容纳不下全部文档内容时，可用鼠标或键盘滚动文档。在文档窗口右侧和底部可设置滚动条，使用滚动条能在文档中快速移动。

键盘两边的【Ctrl + Shift】键用于实现英文、多种中文输入方式之间的切换。【Ctrl + Space (空格)】键则实现英文与选定中文输入方式间的切换。在进行中英文混排的时候，应该注意区分半角和全角字符的差别。关于这方面的详细情况，可以参见中文 Windows 的有关书籍。

选定、查找与替换文本

修改文档时，常常需要删除、移动或复制某部分内容。而在进行这些操作之前，首先要对

那部分内容进行选定(或选取、选亮)。常用的选定方法是按住鼠标左按钮并且将鼠标拖过要选定的内容,使其黑白反色。如果用键盘选定文本,那么应该先按住【Shift】键,然后再按光标移动键或翻页键进行选定。

内容选定之后,或可按【Delete】键将它删除;或对它们进行剪切、复制后,将它们移动到本文档其他地方或其他文档或其他 Windows 应用程序中去(详见下小节)。

在【编辑】栏的下拉菜单中有【查找】和【替换】两个功能。它们不但可用于查看或替换文档中的字符、词组、语句及图形,而且可以查找及替换如域、制表符、分页符、可选连字符以及段落标记等特殊字符。在 Word 6 0 中进行查找或替换的"搜索"方式也十分灵活,可以全程搜索,也可以只向上或只向下搜索。

剪切、移动和复制文本

工具栏中有:剪切按钮;复制按钮;粘贴按钮。

对于被选定的内容,若单击剪切按钮,该内容就从窗口消失,被 Word 移到剪贴板中;对于被选定的内容,若单击复制按钮,则该选定内容被复制到剪贴板中,原窗口内容并不消失。

剪切和复制也可以分别通过选取【编辑】菜单中的【剪切】和【复制】项完成。

此剪切或复制下来的内容可以被移动粘贴。具体方法是:将光标移至待插入位置,然后单击粘贴按钮,被剪下的内容就会重新出现在新位置上。当然,通过选取【编辑】菜单中的【粘贴】命令也能完成相同的工作。

在短距离内移动文本或图形最为方便的方法是:用鼠标进行拖放式操作。首先选定文本或图形,将鼠标指针指向被选定的内容,然后按住鼠标左键并移动,待虚线光标到达待插入位置,松开鼠标按钮便可。短距离内复制文本或图形的操作方法基本与上相同。只是在移动鼠标时,应同时按住【Ctrl】键。

6.2.2 文档的排版

字符格式

字符是指字母、空格、标点、数字及其他符号(如 @、* 、&)。在 Word 中,可以对字符进行字型、大小和样式(粗体、斜体、下划线)等格式化设置。

工具栏上有关字符格式的按钮如图 6 2-2 所示。



图 6 2-2 工具栏上的字符格式按钮

在字符类型列表中,有许多种英文字体可供选择。但一般配置的中文 Word 6 0 只提供宋体和黑体两种汉字,更多的汉字字体选择另需配置字库。

在选定字符后,通过选中上述列表栏目或单击上述按钮,是改变选定字符属性或格式最简

捷的方法。改变字符属性和格式的另一种较为完善的方法是,通过【格式】下拉菜单中的【字体】选项进行。

段落格式

文档段落格式包括:对齐方式、缩进方式、行间距、段间距、制表位、底纹、项目符号和编号方式。

Word 中有四种文字对齐方式:左对齐、右对齐、中间对齐和两端对齐。

要改变某一个段落的对齐方式时,先将插入点移到该段落,然后单击格式栏上所需的段落对齐方式按钮,便可达到目的。通过【格式】下拉菜单中的【段落】选项(见图 6.2-3),不仅可以定义对齐方式,而且可以更细致地对缩进方式、行距、段距、制表位、底纹、项目符号和编号方式等进行定义。

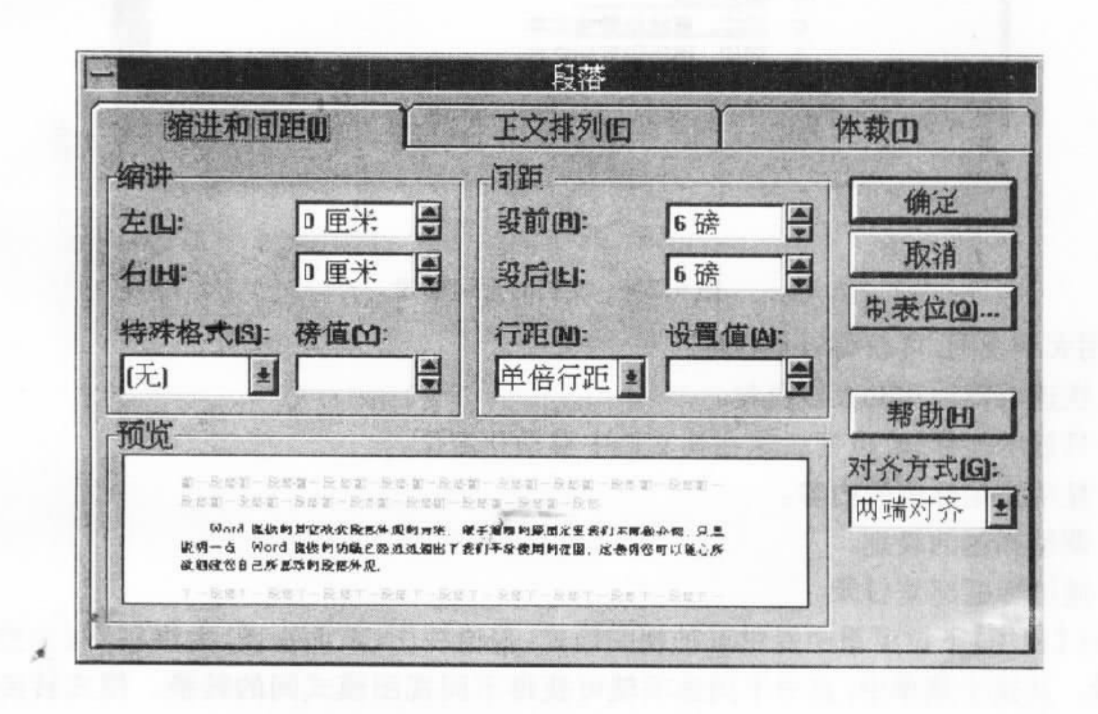


图 6.2-3 更改段落格式对话框

在 Word 中,段落是文字、图形、对象(如公式、图像)及其他项目的集合。段落的最后有一个段落标记。它在屏幕上是否显示可由工具栏上“显示/隐藏”按钮控制。

段落标记不仅标识一个段落的结束,而且还携带该段落格式的定义。删除一个段落标记,那么该段落将和下段合并,该段原先的格式编排将被取消,改用下一段落的格式。为此,建议用户在工作中最好显示出段落标记,以免误删段落标记。

大纲模式

在 Word 中,大纲是指一种视图类型。大纲视图不仅显示文档内容,而且显示文档的结构。这种大纲视图可用于文档创建、修改、重组及校对的各个工作环节。图 6.2-4 是大纲模式下的典型窗口布置。

图中显示了本章的四级标题。章、大节、节、小节的标题级别依次递减,最低一级总是正

文。标题前若以空心“+”号引导，则表示此标题下包含较低级别的内容。

在图6.2-4中工具栏的最下一排按钮是用来操纵标题级别的升降和显示的。

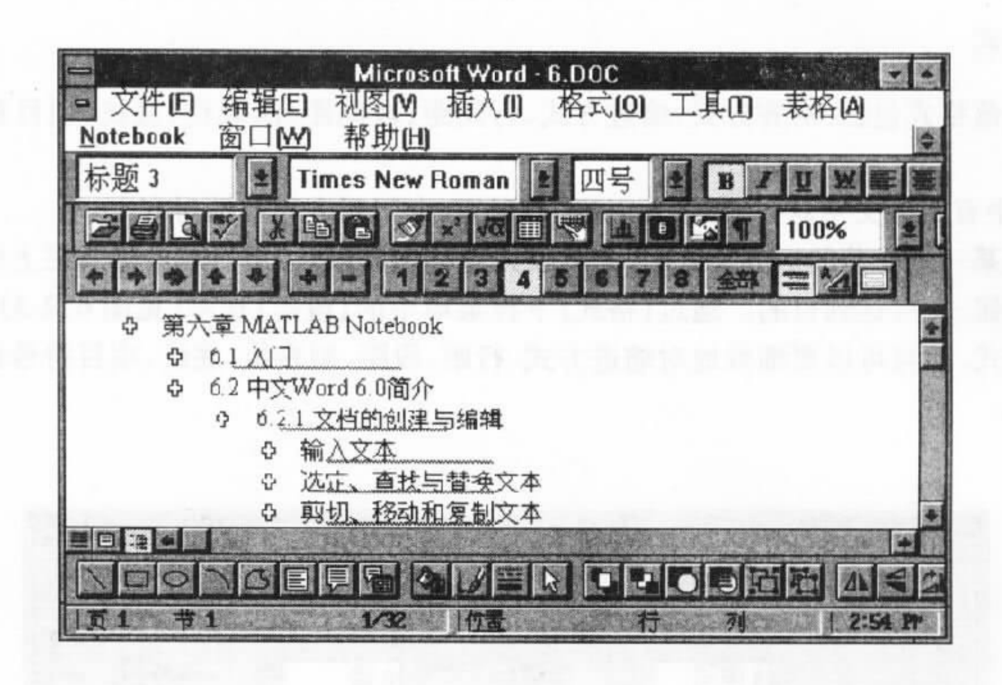


图6.2-4　文档的大纲模式

采用大纲视图，可获得以下好处：

(1) 快速准确地移动大段资料。

(2) 只显示大标题，以便快速在长文档中移动和定位。

(3) 显示指定标题的内容。

(4) 调整标题的级别。

(5) 通过标题建立目录。

Word【视图】下拉菜单中提供五种视图模式：普通视图、页面视图、大纲视图、主控文档和全屏显示。从这个菜单中，点中不同选项就可获得不同视图模式间的转换。模式转换更简捷的方法是用鼠标单击“状态栏”中的按钮(见图6.2-4窗口最下方第二排左端的那三个按钮)。

通过在普通视图、页面视图与大纲视图之间进行切换，用户可在普通视图中细致地编写文档，在页面视图中观看文档的真实排版效果，在大纲视图中进行标题或整节内容的特殊操作。

6.2.3　样式与模板

样式

使用Word样式，可以轻松快速地编排具有统一格式的段落以及文章。Word还允许用户根据需要修改Word中的标准样式。

在开始输入新文档时，Word将用默认“正文”样式来设定文字格式。在创建页眉、脚注和目录等元素时，可应用对所有Word文档均有效的内部标题样式中的一种(如标题1到标题4、Input、Output、Calc、Error等)。当然也可以修改原有内部样式，创建用户自己的样式。

通常，一篇文章总有格式相同的段落。例如，所有二级标题都使用同样的字体、磅值、对齐方式和段间距。在这种情况下，用户完全不必对逐个二级标题分别进行定义，而只需将“标题2”的样式进行适当的修改。例如，将所有二级标题定义为黑体的具体步骤如下：

(1) 选取【格式】菜单中的【样式】选项。

(2) 当“样式”对话框弹出后，在【样式】选择框中选择【标题 2】，然后选择【更改】。

(3) 当“更改样式”对话框弹出后，选择【格式】中的【字体】项。

(4) 当“字体”定义卡弹出后，在中文字体列表中放弃默认的宋体，而改选黑体便可。

模板

模板与样式相似，但样式仅对段落格式进行统一设定，而模板是对整篇文档的格式进行设定。

完全安装的 Word 备有二十几种预先设定好格式的模板。它们均为最常用的各种文档格式。当用户创建新文档时，只需从中挑选适合自己的一个模板，Word 将把该模板的样式和其它要素复制到用户文档中。

模板的定义内容十分丰富。除了文档样式定义外，模板还包括页面设置、图文集、宏命令以及文本内容等定义。事实上，MATLAB Notebook 就是借助专门设计的 M-book 模板而变“活”的。

当需要使用某特定模板来建立文档时，必须调用【文件】菜单上的【新建】命令指定模板。如果用户不在“新建”对话窗中指定所需的模板，Word 将把新文档建立在默认的“Normal”文档模板上。

6.2.4　图文框

图文框可以包括除了脚注、尾注及批注之外的任何文档元素。图文框可放置在页面的任意位置。图文框的作用是：

(1) 让某些段落或项目集中成为整体(如图形、题注)。

(2) 使文本环绕带图文框的项目(如报纸或小册子中的图形镶嵌)。

(3) 在文档任意位置，插放侧标、注释、小图形和其他项目。

图文框被选定(用鼠标点击)后，可以被鼠标拖放到页面的任意位置，也可以通过键入坐标精确定位在页面的某位置上。

假如把随图文框出现的锚形锁定符放置在某段落内，那么这图文框将始终保持与该段落的相对位置不变，并出现在同一页上。

建议用户在页面视图模式下使用图文框。此时，用户能看到图文框在页面上的真实位置。在这种模式下，(借助鼠标实现)图文框大小的调整、图文框位置的移动，都比较真实直观。

插入图文框

插入图文框有两种方法：

(1) 单击工具栏上的“插入图文框”按钮。

(2) 调用【插入】菜单中的【图文框】命令。

插入图文框时，必须在页面视图下。因为只有在页面视图下，才能准确地观察到图文框的位置。

若在文档中选定文字、图形或表格等元素后，再进行插入，那么图文框将把这些选定内容包括在内，图文框的大小会自动设置。

当插入点位于图文框中的时候，可以对图文框中的文字内容进行编辑。把插入点移出图文框后。图文框可以作为一个整体对象被移动和改变大小。

移动图文框

当插入点位于图文框外时，将鼠标指向图文框，按下鼠标，光标指针将变为四向箭头。此时拖动鼠标，一个虚线图文框将随之移动。将虚线图文框移到用户需要的位置，松开鼠标便可。

调整图文框大小

用鼠标调整图文框的大小是最方便的。将鼠标指向图文框，单击鼠标，图文框四周就会出现八个尺寸控制(小方黑)点。使用鼠标拖动尺寸控制点，就可以调整图文框的大小。

图文框与文字的混排

前面讲了图文框位置和大小的鼠标控制方法，至于图文框是否被正文环绕、图文框是否与某指定的段落相连、图文框是否固定于页面的某一个特定位置，都要借助“图文框”对话窗加以确定。具体方法是：将插入点移动到图文框内，调用【格式】下拉菜单中的【图文框】命令，便出现图 6.2-5 所示的对话窗，然后可以进行相应的操作。

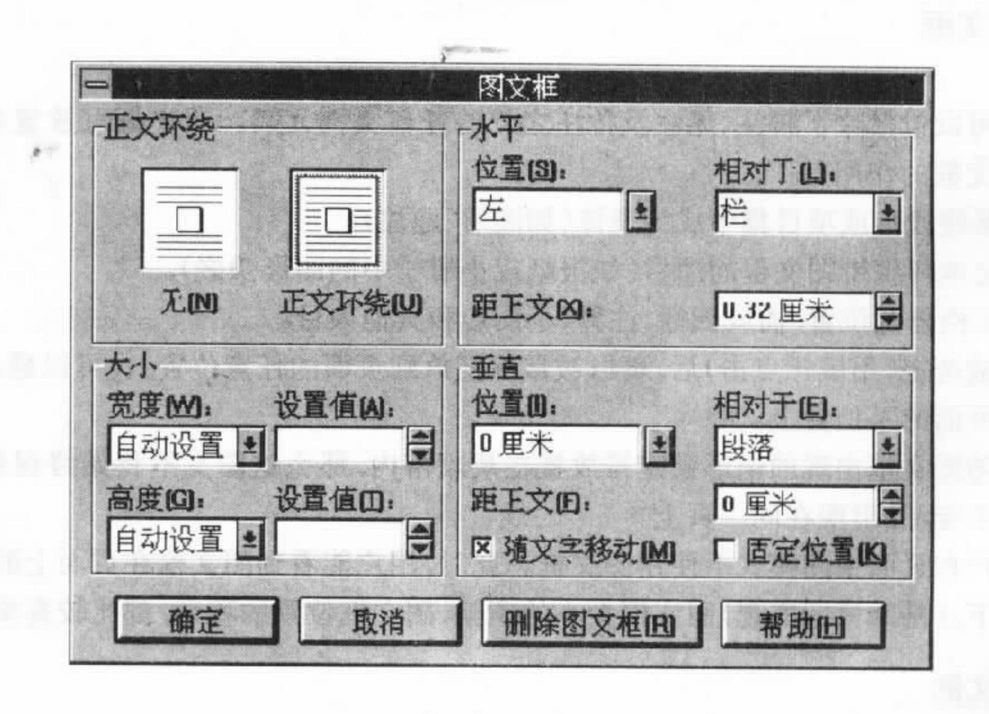


图 6.2-5 设置图文框格式

给插图添加题注

在文档中的图形、表格、公式等项目通常需要添加编号及题注。添加工作既可以人工进行,也可借用 Word 的题注功能自动实现。后一种方法的优点在于:采用自动题注的项目在今后的移动、添加或删除中,都不会造成编号错误,Word 能自动重编。此外,带自动题注的项目可以被交叉引用。添加题注的方法如下:

(1) 选定要添加题注的项目(不仅限于插图)。

(2) 从【插入】菜单中选择【题注】命令,在弹出的“题注”对话框中,适当填写和选择项目的标签和编号。

(3) 添加新的题注。

6.2.5 Word 工作的自动化

向导

在前面已经说过,Word 中所有文档都是基于某种模板建立的。Word 6 0 所提供的模板中有若干类带有“向导”。

“向导”是 Word 提供的一种功能。它通过问答,按用户自己所需自动创建文档格式。使用向导是创建信函、备忘录、摘要、新闻简报以及在文档中插入表格最快的方法。

启动向导的方法:

(1) 从【文件】中选择【新建】命令。

(2) 选定用户所需那种文档的“向导”栏,然后选择【确定】按钮。

(3) 按自己需要回答所弹出的对话框中的问题。

(4) 在完成问答后,按【完成】钮即可。

Word 窗口及工作环境的布置

通常在启动 Word 后,用户所看到窗口和得到的工作环境是 Word 的默认设置。Word 允许用户根据需要重新设计窗口菜单命令、工具栏、键盘设定和对话框默认设置。以下是几种修改设置的常用方式:

(1) 在【视图】下拉菜单中选择【工具栏】,出现“工具栏”对话框。通过选中或取消某列表内容,使 Word 窗口工具栏中的工具类型发生改变。

(2) 在【视图】下拉菜单中选择【工具栏】,出现“工具栏”对话框,再选择【自定义】,便出现如图 6 2-6 所示的三张“自定义”卡片:“工具栏”、“菜单”、“键盘”。

获得这三张卡片的另一条途径是:在【工具】下拉菜单中选择【自定义】。

例如用户在“工具栏”卡片的分类栏中选择“插入”行,于是在其右边出现相应的各种工具按钮。又假如点中√α“插入数学公式”按键,于是在“工具栏”卡片的左下方的说明栏中出现关于此按键的简单说明。假如用户今后常用此按键,则可用鼠标点住此键,并把它拖到 Word 窗口的适当位置便可。

若用户想删去 Word 窗口中的某工具按键,只要用鼠标点住那键,并拖到“工具栏”卡片

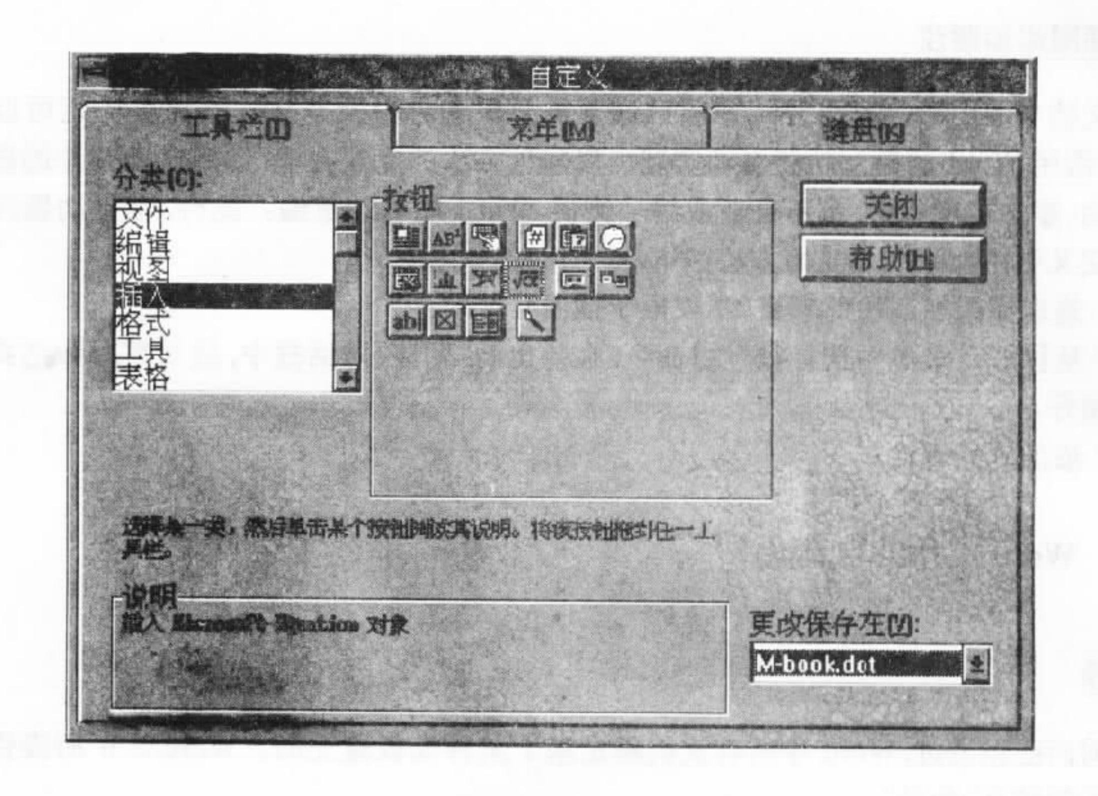


图 6.2-6 自定义卡片

上便可。

(3) 在如图 6.2-6 所示的“菜单”自定义卡片上，可以对 Word 窗口菜单的内容、名称、排列及下拉菜单的栏目进行重设置。

(4) 在如图 6.2-6 所示的“键盘”自定义卡片上，可以对实现各种指令的快捷键进行重设置。

(5) 在【工具】下拉菜单中选择【选项】栏，便会出现“选项”卡片：“常规”、“视图”、“编辑”、“保存”等。在每一张分类卡片上，可实现对 Word 的重设置。

宏

宏就是能组织到一起作为一独立的命令使用的一系列 Word 命令，它能使日常工作变得更容易。实际上，在 Word 窗口上的菜单命令、工具按键等都是相应宏的名字或图标。譬如，“常规”选项卡片就是一个名为“ToolsOptionsGeneral”的宏（通过卡片中选项方式，确定宏所包含的执行指令）。

值得提醒注意：MATLAB Notebook 本身就是应用宏将 Word 与 MATLAB 连接起来的。在运行 M-book 模板后，显示的【Notebook】菜单或 Notebook 工具栏上所获得的命令和操作都是宏命令和宏操作。

在 Word 中宏指令是用 WordBasic 来完成的。WordBasic 与 Visual Basic 类似，是一个在 Windows 环境下的开发工具。使用 WordBasic 可以开发自己的宏指令，也可以编制 Word 与支持 DDE 功能的 Windows 应用程序之间进行通讯的程序。

Word 使用 WordBasic 将宏作为一系列指令来录制。可以在宏编辑窗口中打开录制的宏以修改这些指令，也可以写一些灵活而有效的宏，它包含用户不能录制的 WordBasic 指令。有

兴趣的读者可以自行用 Word 的宏编辑器看看 m-book dot 模板中各种宏指令(包括与 MATLAB 进行通讯的指令)是如何实现的。有关 WordBasic 语言的详细情况,请参阅有关用 Microsoft Word 编程的书籍。

将宏指定到工具栏、菜单或快捷键后,这样用起宏来就像标准 Word 命令一样方便。宏的典型的用法:

(1) 加速日常的编辑和格式工作。

(2) 合并多个命令。

(3) 使对话框中的选项更易于访问。

(4) 使一系列复杂的任务自动执行。

一旦将宏指定到工具栏、菜单或快捷键,运行宏就变得很简单。只需单击相应的工具栏按钮,选择正确的菜单命令或按下对应的组合键。当然也可以从【工具】菜单中选择【宏】命令来运行。

可以使用样式"管理器"对话框来移动、复制或重命名一个宏。在默认情况下,Word 将宏保存在共用模板(Normal dot)中,这样它们对每一个 Word 文档来说都是可用的。然而,如果保存在共用模板中的宏仅对某一特定类型的文档有用,可能需要将宏移动或复制到那个类型文档的模板中去。

6.3 Notebook 的运行环境

6.3.1 Notebook 的安装

MATLAB Notebook 是一个相对独立的部分,其安装程序也是以压缩格式存储在磁盘上,共 1 张高密软盘。安装程序应作为应用程序安装到 Windows 系统上,因此必须在 Windows 环境下安装。MATLAB Notebook 的安装与其他 Windows 应用程序的安装完全相同,安装方法就不再介绍,有关的详细情况请读者参考 MS-Windows 的用户手册。下面我们说明安装时应注意的事项。

(1) 安装 MATLAB Notebook 必须先在用户的计算机系统中安装有:MATLAB 4 2 版的基本核心部分;中文或西文的 Word 6 0 版本。

(2) 当安装程序运行时,在弹出对话框后,按提示输入 MATLAB 系统所在的盘名和目录、Word 6 0 所在的盘名和目录。

(3) 在以上两步结束后,会在 MATLAB 的目录结构中建立 Notebook 子目录。然后,把"matlab \ notebook"目录下的 m-book dot 文件复制(或移动)到"winword \ template"目录下。

安装结束后,在 MATLAB FOR WINDOWS 的程序组中将出现 MATLAB Notebook 程序项,并看到该程序项的图标(见图 6 3-1)。这样,用鼠标单击【MATLAB Notebook Readme】图标就可以启动 MATLAB Notebook,并打开 Notebook 的 readme 文档。

在安装过程中,会自动对 Windows 目录下的 Word 6 0 初始化文件 winword6 ini 进行修改,加入 Notebook 运行时所必须的环境信息。注意:因为中文 Word 的初始化文件为 word6 ini, Notebook 没有考虑与中文 Word 完全兼容的问题,所以 Notebook 会新建一个 windord6 ini

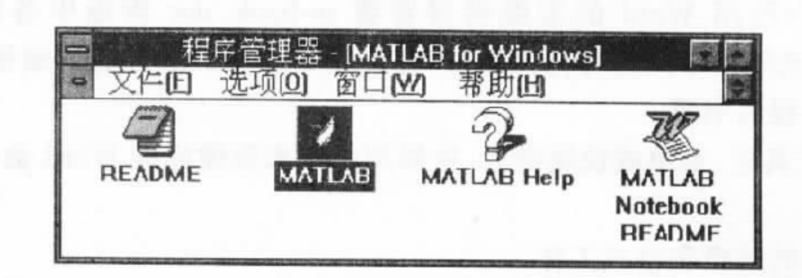


图 6.3-1 MATLAB for Windows 程序组

文件。由于在启动中文 Word 时并不运行 winword6.ini, 因此如果想在中文 Word 中更好地应用 Notebook, 应该将 MATLAB 在 winword6.ini 中建立的各种环境参数加入到 word6.ini 中。

如果程序不是靠安装软件"setup.exe"自动生成, 而是拷贝上去的, 那么应在 MATLAB 目录中建立 Notebook 子目录, 然后将文件逐一拷入该目录, 并在 Windows 目录下的 winword.ini 中的[MATLAB NOTEBOOK]组中增补如下内容:

MATLAB-PATH=C:\MATLAB\bin

NOTEBOOK-PATH=C:\MATLAB\notebook

WINTOOLS-PATH=C:\MATLAB\toolbox\wintools

对于中文 Word 6.0, 应在 word6.ini 中加入如上内容。

6.3.2 启动 Notebook

启动 Notebook(即 M-book)有两种方法: 一是从 Word 中启动 M-book, 一个是从 MATLAB 指令窗中启动 M-book, 详述如下。

从 Word 中启动 Notebook

从 Word 中启动 Notebook 有两个方法:

(1) 在 Word 默认窗口下创建 M-book

假如用户在 Windows 下先打开 Word, 这时 Word 窗口的默认模板是常用的 Normal.dot。在这种情况下, 创建 M-book 的步骤是: 从 Word 窗口的【文件】下拉菜单中选择【新建】子项; 在弹出的对话框中, 调节滑动键, 单击选择"M-book"模板, 按【确定】键; 于是, Word 的窗口布置由原先的默认式样变成"M-book"式样; 假如此前 MATLAB 尚未启动, 则 MATLAB 会自动被启动, 用户可看到 MATLAB 的启动图标; MATLAB 启动结束后, 便进入新的 M-book 文档。

值得注意的是: 如果用户的 Word 6.0 是中文版, 那么在进入 M-book 文档的最后阶段, 将出现图 6.3-2 所示的提示, 用户只管单击【确定】按键便可。这是由于中文 Word 6.0 中没有"ToolOptionsAutoFormat"命令的缘故, 而英文 Word 6.0 中有此命令。本书作者至今尚未发现 Word 中英文版的这种差别对 MATLAB Notebook 的性能产生什么影响, 只是在运行图形指令时, 应先执行一下【Notebook】菜单中的【Notebook Options】命令, 以保证按默认的设置输出图形。事实上读者可以自己建立一个宏指令来完成这个功能。

(2) 在 Word"M-book"窗口下创建 M-book



图 6.3-2 中文 Word 6.0 下特有的提示

如果当前 Word 已经工作在"M-book"模板下，那么 Word 的窗口布置将与默认状态不同。在这种情况下创建新 M-book 文档的步骤是：在【文件】下拉菜单中选择【New M-book】选项便可。

(3) 在 Word 默认窗口下打开已有的 M-book 文件

在 Word 默认的窗口下打开已有 M-book 文件的方法与打开一般文件没有两样。最常用的方法是从【文件】下拉菜单中选择【打开】项，然后从弹出的对话框中选择所需要编辑的 M-book 文件。以后的步骤与方法(1)相同。至于其他打开文件的方法，请读者参考有关的介绍 Word 6.0 的书籍。

从 MATLAB 中启动 Notebook

从 MATLAB 启动 Notebook 比较简单，只需在指令窗中键入 notebook 指令，然后按回车键就可以了。为了更好地理解该指令的实质与用法，读一下 notebook.m 文件是有帮助的。

notebook.m

```
function notebook(fileName)
% NOTEBOOK Open the MATLAB Notebook
%       NOTEBOOK foo opens the M-book 'foo.doc' in the Notebook
%
%       NOTEBOOK, by itself, opens the Notebook for a new
%       M-book called 'document 1'
%
%       See also NBHELP, EDIT, WORKSPACE and PATHTOOL.
%     Copyright (c) 1984-94 by The MathWorks, Inc.
% get Word's startup path
wordPath = getprofl('Microsoft-Word','PROGRAMDIR','c:\winword','winword6');
% get Word's template path
wordTempPath = getprofl('Microsoft-Word','USER-DOT-
PATH','c:\winword\template','winword6');
if nargin == 0
      % create a new m-book
      eval(['!' wordPath '\winword.exe ' wordTempPath '\m-book.dot
/mNewNotebookFromCmdLine&'])
```

```
else
        % open named m-book
        eval(['!' wordPath '\ wınword exe ' fıleName '&'])
end
```

从 notebook.m 可知，在运行 notebook 指令后，MATLAB 是把 Word 运行文件 wınword exe 作外部命令来运行的。并且，当 notebook 指令后没有输入参量时，MATLAB 指定 m-book.dot 作为 Word 启动时的模板，如果有输入参量，即为打开一个已有的 m-book 文件，从而启动 Notebook。

6.3.3　M-book 模板

在第 6 2 3 节中已经介绍过模板及其应用。如果用户不选用其他模板，Word 将自动把 "normal dot"作为默认模板。

Notebook 的核心是 M-book 模板。M-book 模板为用户提供了在 Word 环境下使用 MATLAB的功能。该模板定义了 Word 与 MATLAB 进行通讯的宏指令、文档样式和工具栏。当调用该模板时，Word 便自动将 M-book 模板中定义的宏装人内存，运行 MATLAB，启动 Notebook 用户界面，以及定义文本的样式和细胞。

同其他 Word 的模板一样，用户既可以修改 M-book 模板原有样式，也可以加入新样式。比如在现成的 M-book 模板中，输人(Input)细胞是绿色的，输出(Output)细胞和自活(AutoInıt)细胞是蓝色的，错误(Error)信息是红色的。如果用户想把输出细胞的颜色变为黑色，那么可用以下操作实现：

(1) 选取【格式】菜单中的【样式】选项。

(2) 当样式对话框弹出后，在【样式】选择框中选择【Output】，然后选择【更改】。

(3) 当更改样式对话框弹出后，选择【添加到模板】选项，然后再选择【格式】中的【字体】项。

(4) 将其中的【颜色】选项改为黑色，按【确定】。

这样以后所有的输出细胞都将是黑色的，除非再次更改它。

6.3.4　Notebook 菜单

M-book 模板窗口菜单栏与默认 Normal 模板不同之一是，增加了【Notebook】菜单项，它提供如下的选项：

Defıne Input Cell	Alt-I	定义输入细胞
Defıne AutoInıt Cell	Alt-A	定义自初始化细胞
Defıne Calc Zone	Alt-Z	定义计算区
Undefıne Output Cells	Alt-U	将细胞转换为文本
Group Cells		定义细胞群
Ungroup Cells		将细胞群转换为单细胞
Hıde Cell Markers	Alt-[	隐藏细胞标志

Evaluate Cell	CTRL + Enter	运行当前细胞或细胞群
Evaluate Calc Zone	Alt + Enter	运行当前计算区
Evaluate M-book	Alt-R	运行 M-book 中所有的细胞
Evaluate Loop	Alt-L	循环运行细胞
Notebook Options	Alt-O	定义输出显示的选项
Path Manager		修改 MATLAB 的搜索路径
Workspace Browser		检查及修改工作内存的变量

M-book 模板窗口与通常不同之二是，在【帮助】下拉菜单中加进了以下三个子选项：

MATLAB Help	Ctrl + F1	MATLAB 的帮助系统
Notebook Help		Notebook 帮助系统
About Notebook		Notebook 版权说明

M-book 模板窗口异常的第三个地方是，在【文件】下拉菜单中增设了【New M-book】。

提请用户注意：在以上新增菜单选项中，定义的快捷键与原先 Word 定义的某些菜单快捷键有冲突。结果，凡发生冲突的快捷键都将按以上新定义发挥作用。要改变这种状况，用户必需修改有关快捷键的命名，方法参见本书 6 2 5 节及有关 Word 的书籍。

6.3.5 Notebook 工具条

装载 M-book 模板后，会新增一个【MATLAB Notebook】工具条，它有四个按钮：

MATLAB Path	MATLAB 的路径管理器。
Workspace View	MATLAB 工作内存浏览器。
MATLAB Help	MATLAB 帮助系统。
Notebook Help	Notebook 帮助系统。

用户可根据需要用鼠标将此工具条移到合适的地方，也可以点击工具条左上角按钮而将它关闭。关于这四种功能将在稍后介绍。

工具条的是否自动出现与 Word 是中文版还是英文版有关。对于英文 Word 6 0，当一个 M-book 被创建或被打开后，工具条会自动显示出来。而对于中文 Word 6 0，工具条不会自动显示。为此，用户需进行如下操作：

(1) 运行【视图】菜单中的【工具栏】命令。

(2) 当如图 6 3-3 所示【工具栏】对话框弹出后，选择【MATLAB Notebook】选项，点击【确定】，便在 Word 窗口显示出【MATLAB Notebook】工具条。

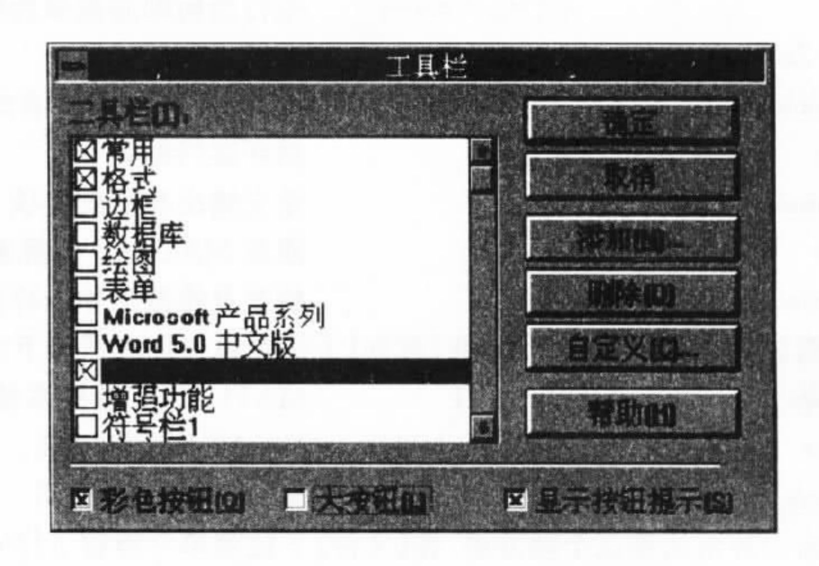


图 6.3-3 定义 MATLAB Notebook 工具条

6.4 Notebook 的运作方法

6.4.1 Notebook 的基本使用方法

Word 文档的输入

Notebook 启动后，可以按 Word 的正常方式输入文本。中文 Word 中默认段落文本的样式是"正文"，英文 Word 中默认段落文本的样式是"Normal"，显示的文字外观采用 Word 中的缺省设置。

MATLAB 命令的输入与运行

在 Notebook 中创建和运行 MATLAB 命令有以下两种方法：

(1) 一步法

像平常输入文本那样，在新的一行里键入一条 MATLAB 命令；按组合键【Ctrl + Enter】，或在【Notebook】下拉菜单中点【Evaluate Cell】。此后，该指令会变成绿色（即成为活的输入细胞），并在其下面行给出运算结果（即输出细胞）。

(2) 二步法

在新的一行里键入一条 MATLAB 命令；在【Notebook】下拉菜单中点【Define Input Cell】选项，则该 MATLAB 命令的字体和颜色都将变成输入细胞样式；再在此后点中【Notebook】下拉菜单中的【Evaluate Cell】，于是在其下给出计算结果。

注意：如果 MATLAB 命令被嵌在一段文本中，那么无论是想直接运行它还是想定义它为输入细胞，都必须先把该命令选亮。否则，抑或出错，抑或把嵌有该 MATLAB 命令的整个段落都定义为输入细胞。

【例 1】产生一个 4 阶 Hilbert 矩阵。

```
hilb(4)
ans =
    1.0000    0.5000    0.3333    0.2500
    0.5000    0.3333    0.2500    0.2000
    0.3333    0.2500    0.2000    0.1667
    0.2500    0.2000    0.1667    0.1429
```

【例 2】将该矩阵有理表示。

```
rats(hilb(4))
ans =
      1       1/2      1/3      1/4
      1/2     1/3      1/4      1/5
      1/3     1/4      1/5      1/6
      1/4     1/5      1/6      1/7
```

说明：在 Notebook 中，MATLAB 命令被执行后所得的结果与在 MATLAB 指令窗中的结果完全一致。在没有输出参量时，输出结果自动赋给预定义变量 ans 。

6.4.2 细胞的使用

在 Notebook 中，MATLAB 命令有两种基本形式：一是输入细胞(Input Cell)；二是自活细胞(AutoInit Cell)。它们都可送到 MATLAB 环境中去运行，所得结果一方面保存在 MATLAB 的工作内存中，另一方面(假如要求的话)送回 Notebook ，运算结果成为输出细胞(Output Cell)。

输入细胞

凡在 MATLAB 环境中合法的命令、注释都可以定义为 Notebook 中的输入细胞。一个输入细胞可以是：一条单独的 MATLAB 命令；在一行内的几条 MATLAB 命令；包括注释在内的多行命令；内嵌在文本中的命令。

输入细胞在 M-book 模板中预定义为：绿色，10 磅大小，"Courier New"粗体英文；作注释的中文是默认段落字体。段落对齐方式、缩进、页面设置等其他格式采用"Normal"模板的设置。如前所述，这些定义是可以更改的。

创建输入细胞和激活的两种方法是：

(1) 输入细胞的创建

不管 MATLAB 命令在独立行还是内嵌在文本中，先选中命令，然后在【Notebook】下拉菜单中点击【Define Input Cell】，于是该 MATLAB 命令的字体和颜色都发生变化。这表明，MATLAB 命令已成为 Notebook 中的输入细胞。这样创建的输入细胞并没有送去运算。

(2) 输入细胞创建与激活的同步实现

将插入点置于独成一行的文本型 MATLAB 命令中，或选亮内嵌在文本中的 MATLAB 命令，选取【Notebook】菜单中的【Evaluate Cell】命令，则不但该命令成为输入细胞，而且被送去计算。运用快捷键【Ctrl＋Enter】操作可替代以上菜单命令的选取。

【例 1】输入细胞的创建。

步骤一：在英文状态下输入以下文本型指令

```
x=0:0.1:2;y=x
```

步骤二：把光标放在上述命令中，或用鼠标把它们选亮。

步骤三：在【Notebook】下拉菜单中选点【Define Input Cell】，于是那文本型命令就成为以下形式的输入细胞：

```
x=0:0 1:2;y=x
```

说明：

(1)【Define Input Cell】的功能仅仅是把文本命令变成输入细胞。

(2) 若要把输入细胞送去计算，则可用【Notebook】下拉菜单中的【Evaluate Cell】命令实现，并产生以下结果：

```
y =
  Columns 1 through 7
         0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000
  Columns 8 through 14
    0.7000    0.8000    0.9000    1.0000    1.1000    1.2000    1.3000
  Columns 15 through 21
    1.4000    1.5000    1.6000    1.7000    1.8000    1.9000    2.0000
```

【例 2】输入细胞创建与激活的同步实现。

先以文本方式键入以下(第一行)命令，然后按【Ctrl-Enter】键，得到以下形式的输入、输出细胞：

```
z=x.^2, v=sqrt(x)
z =
Columns 1 through 7
         0    0.0100    0.0400    0.0900    0.1600    0.2500    0.3600
Columns 8 through 14
    0.4900    0.6400    0.8100    1.0000    1.2100    1.4400    1.6900
Columns 15 through 21
    1.9600    2.2500    2.5600    2.8900    3.2400    3.6100    4.0000
v =
Columns 1 through 7
         0    0.3162    0.4472    0.5477    0.6325    0.7071    0.7746
Columns 8 through 14
    0.8367    0.8944    0.9487    1.0000    1.0488    1.0954    1.1402
```

```
Columns 15 through 21
    1.1832  1.2247  1.2649  1.3038  1.3416  1.3784   1.4142
```

说明：

(1) 该例中变量 x 取自 MATLAB 工作内存。而变量 x 的保存是由上例最后的【Evaluate Cell】计算指令完成的。

(2) 在 Notebook 中，是否显示计算结果的控制方法与在 MATLAB 指令窗中一样。命令后有分号"；"，则计算结果不会以输出细胞显示。命令后有逗号"，"或不带分号，则显示。

自活细胞

自活细胞与输入细胞功能的唯一不同是：当用户启动一个 M-book 文件时，包含在该文件中的自活细胞会自动被送去运算。输入细胞不具备这种功能。若用户需要在打开文件时，对 MATLAB 工作内存进行初始化工作，那么自活细胞特别有用。

自活细胞在 M-book 模板中预定义为：深蓝色，10 磅大小，"Courier New"英文粗体。

自活细胞有两种来源：文本形式的 MATLAB 命令；已经存在的输入细胞。为把它们变成自活细胞，先用鼠标选亮它们，然后运行【Notebook】菜单中的【Define AutoInit Cell】命令即可。

在此要强调指出：在文件启动以后，新定义的自活细胞并不会自动运算，须另外进行运算操作。运行自活细胞的方法同输入细胞一样，选择【Evaluate Cell】菜单命令或按【Ctrl + Enter】键。

输出细胞

输出细胞包含 MATLAB 的输出结果：数据、图形、错误信息。数据的输出样式在 M-book 模板中定义为蓝色，10 磅大小，Courier New 英文细体；错误信息的输出样式是红色，10 磅大小，Courier New 英文粗体，图形的输出格式则通过【Notebook】菜单中的【Notebook Options】来设置，关于此稍后详述。

输出细胞是输入细胞或细胞群运算后产生的，而不是人为定义的。输出细胞总是紧跟在产生它的输入细胞或细胞群之后。假如，已带有输出细胞的输入细胞经修改后重新运行，那么将用新的输出细胞替换原有的输出细胞。

【例 3】输出细胞的产生和修改。

(1)键入以下命令，并运行，于是得输出细胞

```
A=[1,2,3,4,5,6;7,8,0],inv(A)
A =
      1      2      3
      4      5      6
      7      8      0
ans =
   -1.7778     0.8889     -0.1111
    1.5556    -0.7778      0.2222
   -0.1111     0.2222     -0.1111
```

(2) 修改上面输入细胞矩阵 A，再运行，则新的输出结果会覆盖掉原来的输出结果。由于这种修改运行操作只能在屏幕上表现，因此在书面上只能把修改运行后的最终结果重写如下：

```
A=[9,2,3;4,5,6;7,8,0],inv(A)
A =
      9      2      3
      4      5      6
      7      8      0
ans =
   0.1345     -0.0672     0.0084
  -0.1176      0.0588     0.1176
   0.0084      0.1625    -0.1036
```

【例 4】输出细胞位置的比较。

把例 3 中的两条 MATLAB 命令分别定义成两个独立输入细胞，然后运行，那么输出细胞的位置将与例 3 不同。

```
A=[1,2,3;4,5,6;7,8,0]
A =
      1      2      3
      4      5      6
      7      8      0
inv(A)
ans =
   -1.7778      0.8889     -0.1111
    1.5556     -0.7778      0.2222
   -0.1111      0.2222     -0.1111
```

细胞群

细胞群(Cell Groups)是多行输入细胞或自活细胞组成的一个整体。它有三种来源：

(1) 对键入的多行文本型 MATLAB 命令，用鼠标把它们同时选亮，然后在【Notebook】下拉菜单中点选【Define Cell】或【Define AutoInit Cell】，便生成细胞群或自活细胞群。

(2) 对键入的多行文本型 MATLAB 命令，用鼠标把它们同时选亮，然后在【Notebook】下拉菜单中点选【Evaluate Cell】或操作快捷键【Ctrl-Enter】，于是细胞群被创建并激活。

(3) 把已有的多个独立输入细胞或自活细胞同时选亮，然后在【Notebook】下拉菜单中点选【Group Cells】，于是便获得以第一个独立细胞的(自活)性质组合而成的细胞群。

假如被选亮的多个独立细胞区域内夹带有独立成行的文本，那么在把独立细胞组合成群的同时，将把原先夹在中间的文本内容移到细胞群之后。

顺便指出：【Notebook】下拉菜单中的【Group Cells】操作不能直接把文本型 MATLAB 命令组合成细胞群。

细胞群被激活后将拥有一个输出细胞(群)。在这个输出细胞(群)中，输出数据的次序与

命令在(输入)细胞群中的前后次序相同,而图形输出总在数据之后。

细胞群的用途主要有两个:

(1) 为保证 MATLAB 命令结构(如循环结构、条件结构)的完整,必须使用细胞群。

(2) 为保证输出结果(如图形)的完整,必须使用细胞群。

【例5】对循环结构使用细胞群(图6.4-1)。

```
clear
x=0:10;
for k=1:10
    y=k*x;
    plot(x,y)
    hold on
end
hold off
```

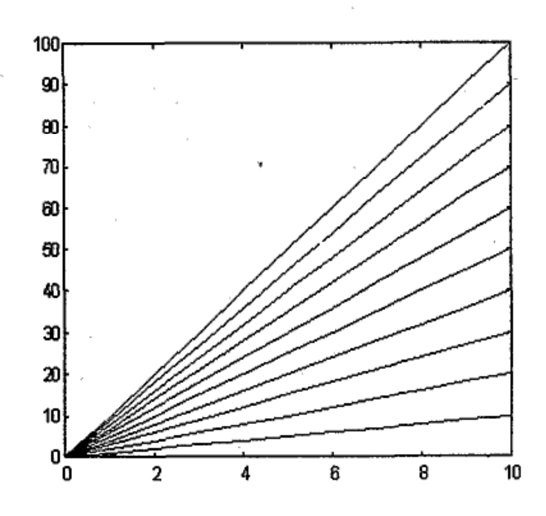


图6.4-1　用循环语句画多线图

说明:假如本例中的细胞群用一行行独立的输入细胞替代,那么当运行它们时,将显示出错警告。

【例6】使用细胞群产生完整图形。

```
clf;
t=0:0.05:10; yt=exp(-t).*cos(t);
tt=fliplr(t); nt=length(t);
la=1.05*ones(1,nt); lb=0.95*ones(1,nt);
et=1-yt;
T=[t,tt]; L=[la,lb];
hold on
```

```
fill(T,L,'y');          % 把纵坐标 0 95 到 1 05 涂成黄色
plot(t,et,'r');
plot(t,la,'y',t,lb,'y');
ts=t(max(find(abs(1-et)>0.05)))
set(gca,'box','on'); %坐标轴变成封闭框,gca 是得到当前坐标轴句柄的指令。
hold off;
ts =
    2.9500
```

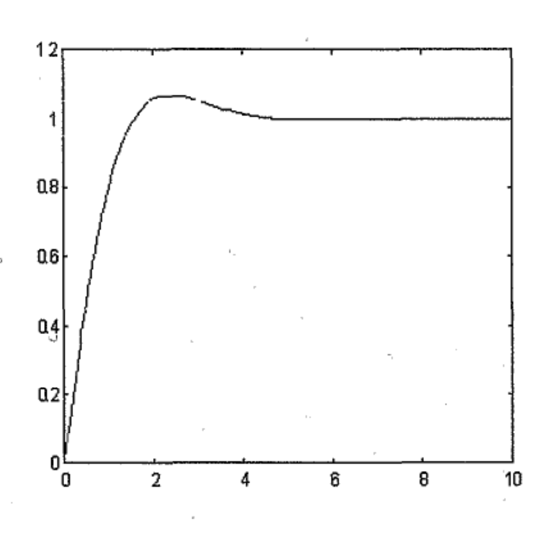


图 6 4-2 使用细胞群创建完整图形

说明:

(1) 假如该细胞群替换为一行行独立的细胞,那么运行后将产生五帧图形。除最后一帧是所需的完整图形以外,其余都是绘制中间的半成品。

(2) 细胞群运行后,不管输出数值结果的 MATLAB 命令在细胞群中的什么位置,细胞群在输出结果时,总是把数值结果放在图形结果的前面。

(3) 要想区分一个细胞群和一组独立的输入细胞,可以使用【Notebook】菜单中的【Show Cell Markers】命令。这个命令将把"整体"标志"[ ]"加在细胞群的首尾两端;而对于一行行单独的输入细胞,"整体"标志将分别显示在每个独立输入细胞的首尾。

(4) 可以用【Notebook】菜单中的【Ungroup Cells】命令将细胞群转化为一组单独的输入细胞。

计算区

计算区(Calc Zones)是一个由文本、输入细胞和输出细胞组成的连续区,用于描述某个具体的作业或问题。在计算区里,用户可以根据描述问题的需要安排段落、标题、格式、分栏,而不受计算区外有关格式定义的束缚。

Notebook 将计算区定义为 Word 的一个节,并将节的分隔符号置于计算区的首尾,但是在文档的开始和结束的地方 Word 不显示节分隔符号。

创建计算区的方法是：先选定包含输入细胞、输出细胞和文本的一个连续区，然后选择【Notebook】中的【Defıne Calc Zone】命令。

要运行计算区，只需将光标置于计算区中的任何位置，然后选择【Evaluate Calc Zone】命令便可。

细胞转化为文本

细胞（包括输入细胞和输出细胞）与文本不同，它们是“活”的。所谓“活”细胞是指当用户改变输入细胞并再次运行时，它的输出细胞也会更新，即新的结果会覆盖掉原来的结果。假如用户希望保存原来的结果，那么就应该将“活”细胞转化为“死”细胞（即文本）。

细胞转化为文本的方法是·选定细胞，运行【Notebook】菜单中的【Undefıne Cells】命令；或将光标置于细胞之中，按组合键【Alt + U】。细胞转化为文本后其样式会改变，由原来的样式改变为“Normal”样式，因此它的字体、大小、颜色均为“Normal”样式。

当某输入细胞或细胞群被转化为文本时，与之相应的输出细胞也被自动转化为文本。然而当输出细胞被转化为文本时，就切断了它与输入细胞的任何联系，输入细胞的性质不会因此而改变。此后，若再次运行输入细胞，Notebook 会紧跟在输入细胞之后产生一个新的输出细胞。

6.4.3 文档中操作 MATLAB 的进一步说明

工作内存的初始化

M-book 的所有计算是在 MATLAB 中进行的，参与运算的所有变量都储存在 MATLAB 工作内存里。各 M-book 文件和 MATLAB 指令窗分享同一个“计算引擎”和同一个工作内存。工作内存中的变量是各 M-book 文件和 MATLAB 指令窗工作后共同产生的。对此，用户应有清醒认识。

当用户同时打开几个 M-book 文件或在 MATLAB 指令窗和 M-book 文件间交互运作时，要特别注意不同文件和窗口之间变量的相互影响。假如要保证某 M-book 文件独占MATLAB 工作内存，保证该文件的输入输出数据间的一致性，一个有效的办法是：把 clear 作为该文件中第一个输入细胞或自活细胞。

【例 1】利用 clear 保证其后程序的工作不受外来变量的影响（图 6 4-3）。

以下这段程序用以表现二阶系统复数极点实部变化对阶跃响应的影响。本例需要 MATLAB 控制工具箱（Control Toolbox）的支持。

```
clear
t=0:0.05:7;
numberofcurves=12;
y=zeros(length(t),numberofcurves);
n=1;
while n<=numberofcurves,
```

```
        [num,den]=zp2tf([],[-n/4+3*i, -n/4-3*i],(n/4)^2+9);
        [y(1:length(t),n),x,tdumb]=step(num,den,t);
        n=n+1;
    end;
    mesh(t,1:numberofcurves,y');
```

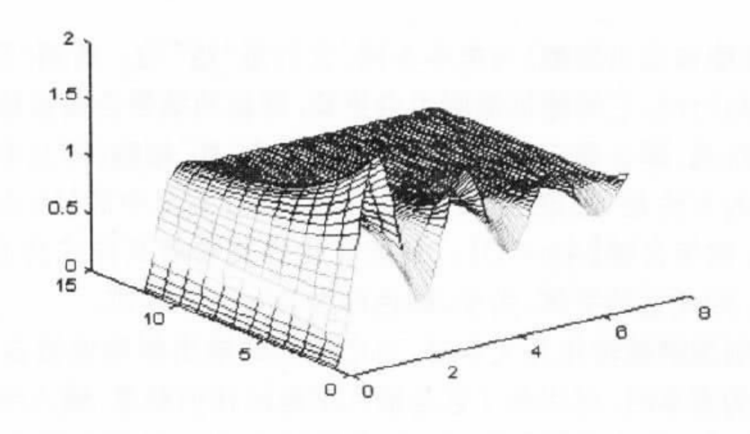


图 6.4-3　clear 在创建新图中的作用

细胞的循环运行

Notebook 提供循环运行细胞的命令。具体步骤如下:

(1) 选择要重复运行的输入细胞,可以包含合法的文字或输出细胞。

(2) 选择【Evaluate Loop】命令,弹出如图 6.4-4 的对话框。

(3) 在"Stop After"栏中输入重复运行次数,然后单击【Start】。Notebook 开始运行命令,并显示已运行的次数。

如果要在每一个循环后加入延迟,可以点击【Slower】按钮;反之点击【Faster】按钮。若要暂停命令的运行,按【Pause】按钮,停止运行命令,选择【Stop】。

注意:本功能在中文 Word 6.0 中无法实现。

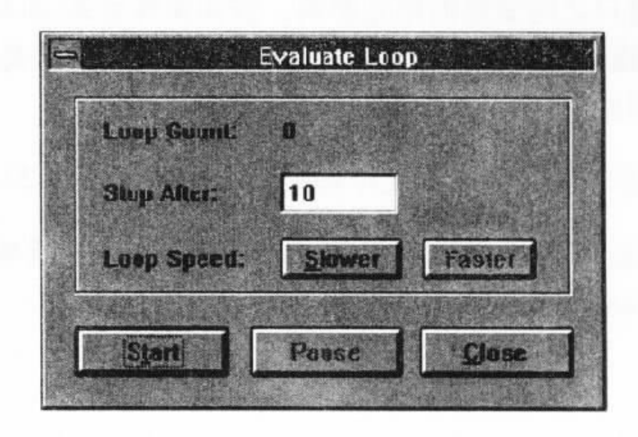


图 6.4-4　循环对话框

【例 2】利用循环运行命令绘制图 6.4-1 。

先运行以下指令：

```
clear;x=0:10;k=1;hold on
```

选亮以下三行指令，打开循环运行对话框，并取重复次数为 10，便可启动。

待重复次数完成后，关闭循环运行对话框，图形便绘制结束。

```
y=k*x;
plot(x,y)
k=k+1;
```

说明：在以上运作结束后，最好再运行"hold off"指令，避免以后图形被重叠。

对 M-book 中全部细胞的操作

在【Notebook】菜单中有两条对 M-book 文件中所有细胞实现整体操作的指令：一条是【Evaluate M-book】；另一条是【Purge Output Cells】。

【Notebook】菜单中的【Evaluate M-book】命令可以运行整个 M-book 文件，即把文档中所有输入细胞送到 MATLAB 中去运行。不管光标处在该文档的什么地方，运行总是从文件首部开始。在整个 M-book 文件运行时，它不但会把所有原输出细胞中的内容刷新；而且会补写新的输出细胞。这个命令在保证整个 M-book 文件中所有指令、数据、图形的一致性方面十分有用。

【Purge Output Cells】命令的作用是删去 M-book 文件中的所有输出细胞。它的具体操作步骤是：在【编辑】下拉菜单中运行【全选】命令，使整个文件选亮，然后再运行【Notebook】下拉菜单中的【Purge Output Cells】命令，所有输出细胞就被删去。这个指令在撰写报告、布置作业时常会用到。

6.4.4 输出控制与文档的打印

数据输出格式

在 Notebook 中，输出数据的格式是通过【Notebook】下拉菜单中的【Notebook Options】选项进行的。该选项运行后，会产生如图 6.4-5 所示的对话框。

(1) 数据格式控制

在【Numeric Format】子选框中的下拉列表中有 8 种可选格式："Short"、"Long"、"Hex"、"Bank"、"Plus"、"Short e"、"Long e"、"Rational"。它们的作用与 MATLAB 指令窗中相应的 format 指令完全一样（详细请看第 3.1 6 节）。

事实上，这 8 种格式也可以通过输入细胞中的 format 命令来控制，其效果相同。

(2) 输出数据间的空行控制

在【Numeric Format】子选框中的【Loose】和【Compact】用来控制输入细胞与输出细胞之间的空白区间。比如，选择【Loose】后，在 M-book 文档的输入细胞和输出细胞之间加入一个空行。

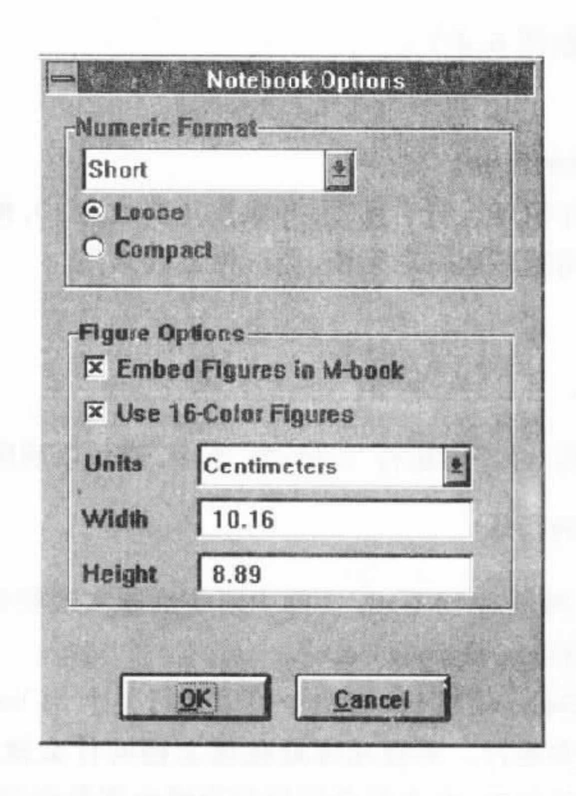


图 6.4-5 输出数据的格式对话框

注意:这种控制方法与输入细胞群中的"format loose"、"format compact"命令有不同的功能。后者控制的将是输出细胞与输出细胞之间空行。

【例 1】试比较几个不同的空行输出格式。

在本例中,数据格式通过【Numeric Format】子选框中"Short"选项实现。

(1) 在【Numeric Format】子选框中选择【Compact】后,运行以下细胞群。在输入细胞和输出细胞之间、输出细胞和输出细胞之间都没有空行。

```
rand('seed',0),A=rand(3)          % 在"0"种子下,生成一个 3 阶随机方阵
B=A*inv(A)                        % A 与 A 的逆相乘生成一个单位阵
A =
    0.2190    0.6793    0.5194
    0.0470    0.9347    0.8310
    0.6789    0.3835    0.0346
B =
    1.0000    0.0000    0.0000
    0.0000    1.0000    0.0000
    0.0000    0.0000    1.0000
```

(2) 假若用户在【Numeric Format】子选框中,选【Loose】后,再运行以上细胞群,那么将能看到在输入细胞和输出细胞间有空行。

(3) 再若用户在【Numeric Format】子选框中,选【compact】后,先运行 format loose 指令,则可看到输出细胞和输出细胞之间有空行。

【例 2】输入细胞中数据格式指令的控制作用。

在本例中，【Numeric Format】子选框中选定“Short”和【Loose】子项不变。

(1)短格式浮点表示

```
format short e
rand('seed',0),A=rand(3)
A =
  2.1896e-001    6.7930e-001    5.1942e-001
  4.7045e-002    9.3469e-001    8.3097e-001
  6.7886e-001    3.8350e-001    3.4572e-002
```

(2) 有理表示

```
format rational
rand('seed',0),A=rand(3)
A =
     589/2690       502/739       2314/4455
      39/829       3850/4119       585/704
     909/1339       265/691       507/14665
```

说明.在此再需强调，不同输出格式给出不同的数据显示精度，但内部存储及运算都是以相同的双精度进行的。

图形输出

如图 6 4-5 的【Notebook Options】对话框也实现对细胞运算输出图形的控制。

(1) 黑白反色的控制

在中文 Word 的情况下，假如在图形输出前，不对【Notebook Options】对话框进行操作，并点选【OK】键；也没在 MATLAB 指令窗中，运行“whitebg”指令，那么在 M-book 文档中细胞运算输出图形将以黑色为背景。这种图在黑白打印机上的硬拷贝将是一片灰黑。

用户若想得到白色背景的图形，必须进行以下两种操作中的一个：打开【Notebook Options】对话框，点选【OK】键；或在 MATLAB 指令窗中，运行“whitebg”指令。

(2) 曲面色彩控制

假如在图形输出前，在【Notebook Options】对话框中不选择【Use 16-Color Figures】(即选择小方块为空白)，那么生成的 256 色曲面图就可能得不到正确的色彩表达，更严重的是在单色打印机上所得到的曲面图将是一片黑色。

(3) 图形镶嵌的控制

M-book 的缺省设置是：在【Notebook Options】对话框中选择【Embed Figures in M-book】(即选择小方块中打叉)。在这种设置下，输出的图形被镶嵌在 M-book 文档中。假如不选择【Embed Figures in M-book】，那么图形将输出到另开的图形窗口中。

(4) 图形大小的控制

在图 6 4-5 所示的【Notebook Options】对话框下方有三个控制图形精确大小的栏目：“Units”、“Width”、“Height”。

当然,所得图形也可用 Word 工具对它进行移动、缩放、剪裁和编辑。读者可以参考前面 Word 的简介或有关 Word 书籍。

【例 3】采用 M-book 缺省设置绘图(图形大小缺省值为:宽 10.16cm、高 8.89cm)(图 6.4-6)。

```
x=0:0.1:4;
y=sin(x.^2).*exp(-x); z=cos(x.^2).*exp(-x);
plot(x,y,'r',x,z,'b');
text(0.65,0.6,'z=cos(x^2)*exp(-x)');
text(1.5,0.25,'y=sin(x^2)*exp(-x)');
title('Plot of sin(x^2)*exp(-x) and cos(x^2)*exp(-x)');
```

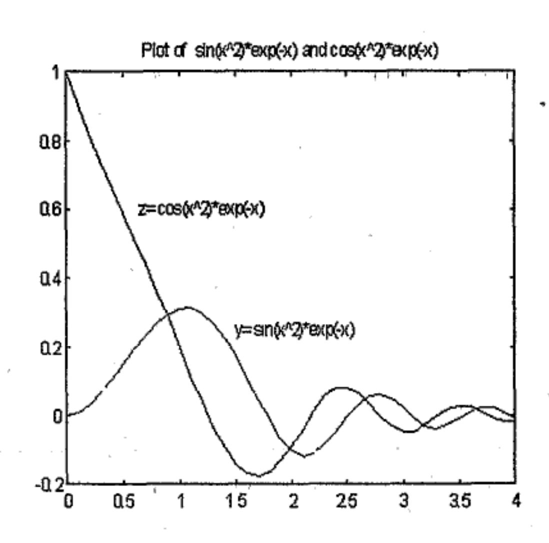


图 6.4-6 采用 M-book 图形大小缺省值所绘的图

【例 4】采用 16 色设置画复数立方函数图(图 6.4-7)。

```
z=cplxgrid(20);
cplxmap(z,z.^3);
title('z^3');
```

说明:MATLAB 的绝大多数图形都能在 M-book 中实现。但动画和某些交互式的图形指令例外,稍后详述。

M-book 文档的打印输出

M-book 的打印输出与其他 Word 文档一样,按标准打印方式进行。当用户在【文件】下拉菜单中选择【打印】命令后,会出现如图 6.4-8 所示的打印对话框。

在这对话框里,可实现以下打印设置:

(1)"打印内容"列表中有:打印摘要信息、批注、样式、自动图文集词条或键值等选项。

(2)"打印范围"子框中有三种选择。

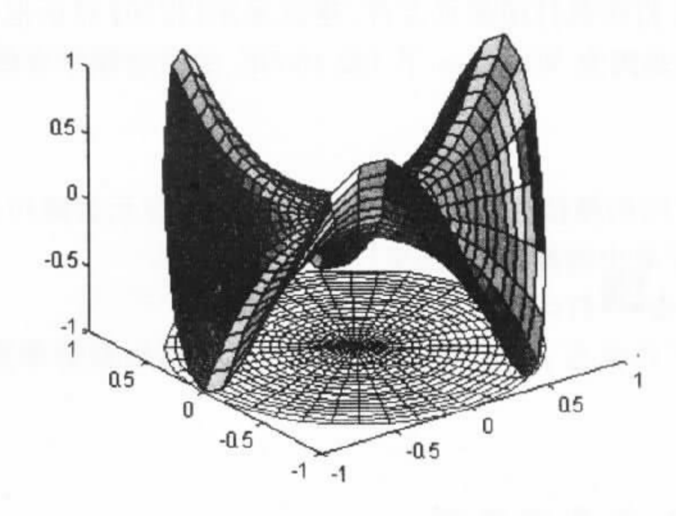


图 6.4-7 采用 16 色设置画复数立方函数图

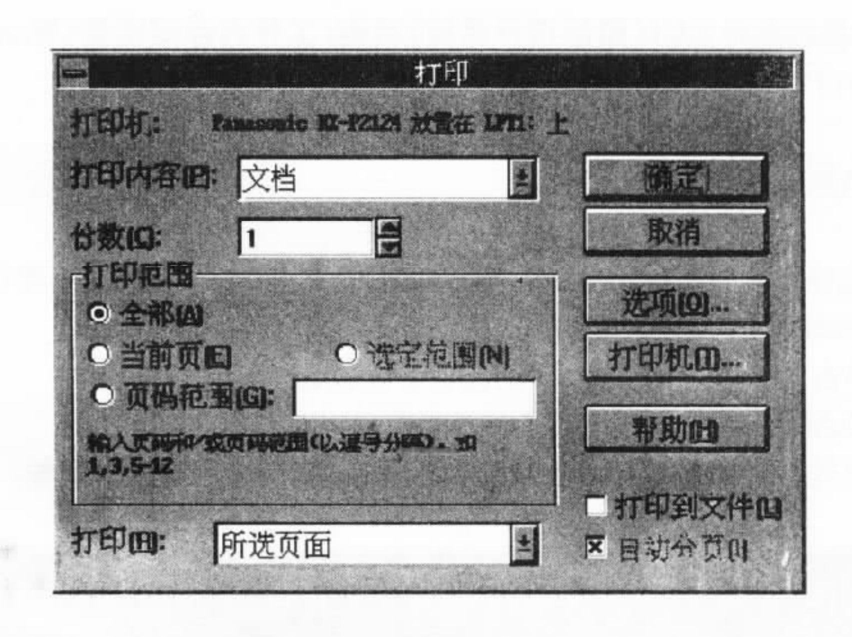


图 6.4-8 打印对话框

(3) 文档备份数目和奇偶页面选择。

(4) 是否以文件形式输出选择。假若在选择框中打叉，则在“打印”对话框中点击【确定】按键后，会出现“打印输出到文件”对话框。在此框里，填写生成文件应在的驱动器名、目录名和文件名，然后点击【确定】按键。这样所得的输出文件可以直接在 DOS 环境下输出到打印机上打印，即便不在 Word 环境下也行。

(5) 点中打印对话框中的【选项】按钮，将可对是否后台打印、送纸方式进行设置。

在缺省安装下，中文 Windows 提供两种 TrueType 汉字字体：宋体和黑体。这种比例缩放字体，使屏幕上显示的与打印纸上的实际结果一致。假如用户想安装其他汉字字体，请参阅有关手册。

最后，还要指出：

(1) 细胞标志(Cell Markers)属于 Word 控制符号，将不被打印出来。

(2) 若用 LQ-1600K 打印机打印多页文件，建议采用【打印】对话框中单页打印设置。否则容易产生打印错误，这是因为 Windows 下 LQ-1600K 的驱动程序有些问题。

打印预览

Word 提供了文档打印的模拟显示功能。实现这种模拟显示的操作主要有以下三种：

(1) 在【文件】下拉菜单中选择【打印预览】项。

(2) 在工具栏中，点击 打印预览按键。

(3) 在"页面视图"工作状态下，调节不同的显示比例，可以获得单页、双页或多页显示效果。

6.5 路径管理器和内存浏览器

Notebook 还提供两种 GUI(图形用户界面)功能：工作内存浏览器(Workspace Browser)和路径管理器(MATLAB Path Manager)。

6.5.1 MATLAB 的路径管理器

路径管理器允许用户检查和修改 MATLAB 的搜索路径。运行路径管理器有三种方法：

(1) 选择【Notebook】菜单中的【Path Manager】命令。

(2) 用鼠标单击 Notebook 工具条中的路径管理按钮。

(3) 从 MATLAB 指令窗中键入命令 pathtool。

当进行以上三种操作中的任何一种后，将出现如图 6.5-1 所示的对话框。

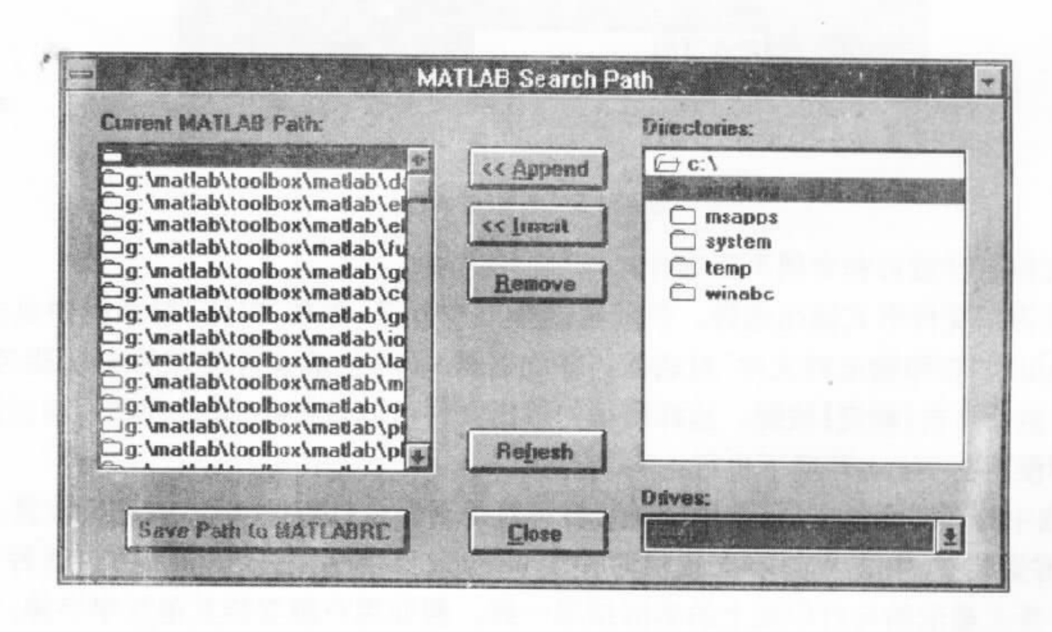


图 6.5-1 MATLAB 路径管理器

在路径管理器对话框中，左栏是当前 MATLAB 路径表，右栏是待加入的目录名选栏，右

下栏是待加目录所在驱动器栏。在对话框中，各按键的功能如下：

(1) Append　　将右栏中所选目录加到搜索路径的尾端。

(2) Insert　　将右栏中所选目录插到搜索路径之中。

(3) Remove　　从搜索路径中删去左栏中所选目录。

(4) Refresh　　观看当前搜索路径。

(5) Save　　把修改了的路径保存到 MATLAB 启动文件 matlabrc.m 中。

6.5.2 工作内存浏览器

工作内存浏览器，有三个功能：列出内存变量（其作用跟 MATLAB 指令窗中的 whos 一样）；删除某些内存变量（其作用如 clear）；检查和更改变量（其作用与 MATLAB 指令窗运行 edit 相同）。

启动内存浏览器的方法有三个：

(1) 运行【Notebook】菜单中的【Workspace Browser】命令。

(2) 单击 Notebook 工具条上的内存浏览器按钮。

(3) 在 MATAB 指令窗中运行 workspace 命令。

启动内存浏览器后，出现如图 6.5-2 所示的对话框。

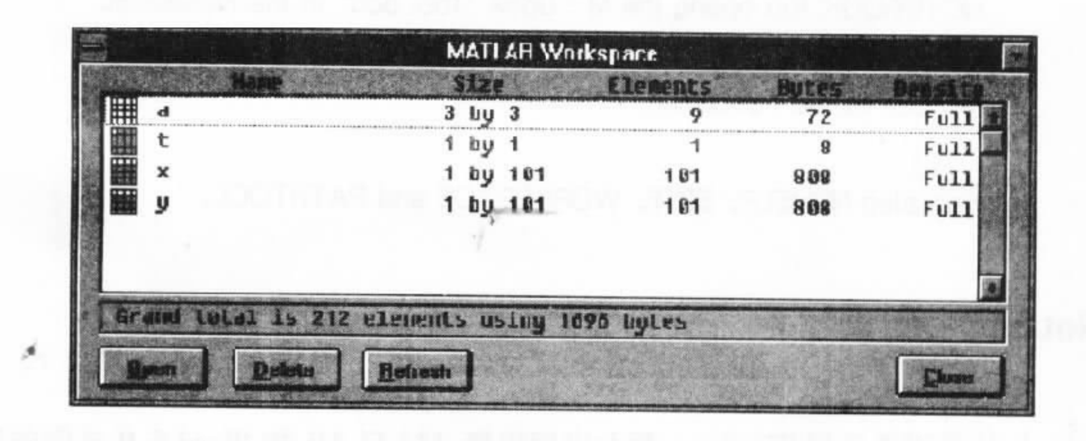


图 6.5-2　MATLAB 工作内存浏览器

对话框列出内存中所有变量的名称、维数、元素数、字节数及是否稀疏。每一行首的方形图标形象表示该变量或是标量，或是向量，或是矩阵。在对话框下方各按键的作用分别是：

(1) Open　　用矩阵编辑器打开所选变量。在那里，可对变量进行修改。

(2) Delete　　从内存中删去某个选定变量。

(3) Refresh　　显示当前内存内容。

关于矩阵编辑器已在第 3.1.2 节中介绍过，这里不再说明。另外，用鼠标双击变量图标也可以打开矩阵编辑器。

6.5.3 Notebook 的帮助系统

在 M-book 模板窗口，同时存在三套帮助系统：Word 帮助系统；MATLAB 帮助系统；Note-

book 帮助系统。它们的存在为用户在统一环境中实现文字、图表、科学运算和计算结果可视化的综合处理提供了有力的在线支持。

启动帮助系统的主要方法有两个：

(1) 打开菜单条上的【帮助】下拉菜单。在此菜单中的前五个选项是关于 Word 的帮助系统;【MATLAB Help】给出关于 MATLAB 主包的帮助;【Notebook Help】专门提供描述 Notebook 特殊功能的帮助。

(2) 在 M-book 模板窗口的工具条上，有三个帮助按钮：

Word 帮助系统启动按钮。

MATLAB 帮助系统启动按钮。

Notebook 帮助系统启动按钮。

事实上，在 Notebook 中，也可以使用 MATLAB 的其他键盘帮助指令，如 help、lookfor、who、whos、which、whatsnew、type 等。由于通过这种方法得到的帮助信息出现在文档窗口，会扰乱文档版面，因此除特殊需要外一般都不用。

【例 1】利用输入细胞调用 help 指令。

```
help notebook
NOTEBOOK Open the MATLAB Notebook
         NOTEBOOK foo opens the M-book 'foo.doc' in the Notebook.
         NOTEBOOK, by itself, opens the Notebook for a new
         M-book called 'document 1'

         See also NBHELP, EDIT, WORKSPACE and PATHTOOL.
```

6.6 Notebook 使用须知

Notebook 的使用涉及到 Windows 两大应用程序：MATLAB 和 Word 6 0 之间的通讯，其中一些较复杂的通讯问题有待解决。本节将对 Notebook 现行版本中的一些问题及使用中应该注意的问题给予说明。

6.6.1 Notebook 现行版本问题

现行版本 Notebook 的问题表现在两方面：一是与 Word 的冲突；二是对 MATLAB 功能包容上的缺陷。

由于 Notebook 是针对英文 Word 开发的，因此，Notebook 拥有英文 Word 的全部功能，且几乎没有一点冲突。本文作者使用至今，仅发现三个快捷键操作：【Alt-A】、【Alt-I】、【Alt-O】由原 Word 中打开下拉菜单【表格】、【插入】、【格式】功能变为 Notebook 中的【Define AutoInit Cell】、【Define Input Cell】、【Notebook Options】功能。用户可以通过“自定义”改变这种冲突。

现行版本的 Notebook 已经包容了 MATLAB 的绝大多数功能，但尚有以下一些功能无法

实现:

(1) 在 Notebook 中无法运行的指令有:交互式指令(或与键盘交互,或与鼠标交互);动画指令;程序调试指令,如 comet、dbclear、dbcont、dbdowninput、dbstep、dbstop、dbtype、dbup、ginput、gtext、input、keyboard、more、movie、pause、zoom 等。当然,包含上述指令的程序也不能在 Notebook 中运行。假如用户计算中必须使用上述指令,那只得在 MATLAB 指令窗中进行,然后把结果送到 Notebook 。

(2) 带 UI 控制和菜单的 MATLAB 图形被嵌入 Notebook 后不起作用。

(3) SIMULINK 不能在 Notebook 运作。

6.6.2 中文版的特殊问题

由于 MATLAB Notebook 是针对西文 Word 开发的,Microsoft 公司仅对 Word 6.0 中文版略作过改动,因此在中文版 Word 6.0 下运行 Notebook 时会有一些问题:

(1) 每新建或打开一个 M-book 文档时,Word 会弹出一个对话框,提示表明宏指令"ToolsOptionsAutoFormat"不存在。用户可以不理会它。

(2) 在中文 Word 6.0 环境下启动 M-book 时,如果出现 Word 要求转换文本的对话框,那么应该关闭"识别英文 Word 中汉字"的功能。方法是:选取【工具】菜单中的【选项】命令,弹出如图 6.6-1 所示【选项】对话框;在【常规】选项中关掉【识别英文 word 文件中的汉字】功能开关。

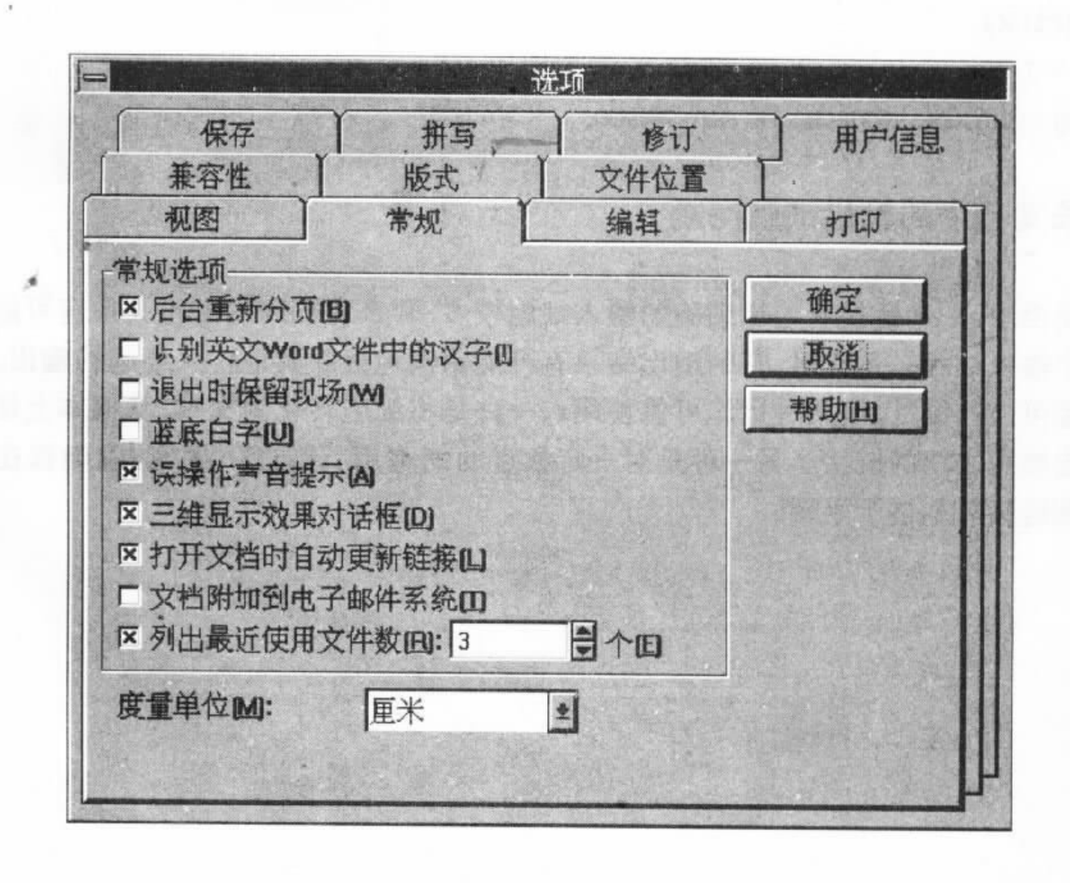


图 6.6-1 Word 中的选项设置

(3) Notebook 工具条不会在中文 Word 中自动弹出。把 Notebook 工具条布置在工具栏的方法见第 6 3 5 节。

(4) 在中文 Word 6 0 情况下，M-book 模板会重新定义一个“Normal”样式。因此，M-book 有两个默认段落样式名称：“正文”和“Normal”。这对使用没有任何影响。

6.6.3 标点符号问题

用户要特别注意中英文标点符号的区别。在输入中文文本时，当然应使用中文标点符号。但要切记：MATLAB 命令、命令组中所用的是英文标点符号。千万不要在 MATLAB 命令、命令组，在输入细胞、细胞群中错误地使用中文标点符号，不然，轻则出错，重则死机。

在标点符号中，尤其要注意冒号、分号、逗号的中英之分。

【例 1】MATLAB 对中文和西文标点符号的反应（由于出版印刷原因，无法从外观上形象表达标点区别）。

```
x=1:4;          % 西文标点符号
y=sqrt(x)
y =
    1.0000    1.4142    1.7321    2.0000
x=1:4;          % 中文标点符号
y=sqrt(x)
??? x=1£
Missing operator, comma, or semi-colon.
```

6.6.4 长文档中的输出细胞问题

在长文档中，如果链接着输出细胞的输入细胞较多，那么它们间的链接关系有可能会被搞乱。当某个输入细胞被运作时，它的输出结果有可能错误地更新其他输入细胞的输出。

为防范可能的混乱，有两种措施可供使用。一种是将细胞转化为文本，从根本上切断输入细胞与输出细胞之间的链接。另一种是对一定数量的细胞用“计算区”界定，以确保在各计算区内的细胞链接关系过于繁杂。

第七章　SIMULINK 动态仿真集成环境

7.1　引　导

SIMULINK 是实现动态系统建模、仿真的一个集成环境。它的存在使 MATLAB 的功能得到进一步的扩展。这种扩展的意义表现在:(1)实现了可视化建模。在 Windows 视窗里,用户通过简单的鼠标操作就可建立起直观的系统模型,并进行仿真;(2)实现了多工作环境间文件互用和数据交换,如 SIMULINK 与 MATLAB, SIMULINK 与 C、FORTRAN, SIMULINK 与 DSP, SIMULINK 与实时硬件工作环境等的信息交换都可以方便地实现;(3)把理论研究和工程实现有机地结合在一起。

本章从 SIMULINK 工具包安装开始,先通过一个简单模型的创建示例向读者展现框图模型创建、仿真的大致过程,进而对主要的操作做更详细的介绍。仿真算法的选择、参数设置以及实施仿真的不同操作方法是第 7.3 节的内容。第 7.4 节是仿真系统的线性化模型。该节介绍如何从非线性框图模型获取平衡工作点和线性化模型,如何从离散-连续混合框图模型获得任意指定采样频率下的等效模型。理解 SIMULINK 的关键在 S-函数。这部分内容比前面几节更深入 SIMULINK 的核心。读者有了前面的基础准备就较容易掌握 S-函数。因此,关于 S-函数的内容被安排在最后一节。本章不介绍与实时硬件有关的内容,这是因为国内很少使用。

7.1.1　系统要求

SIMULINK 的广泛应用是从 MATLAB 4.0 版开始的。在 4.0 版中,SIMULINK 被包含在 MATLAB 的核心执行文件中;在 MATLAB 4.2 版中,SIMULINK 则以 MATLAB 的工具包形式单独出现。因此,SIMULINK 工具包需专门购买和安装。

SIMULINK 运行对系统的要求,除要求增加 2M 硬盘空间外,其余和 MATLAB 完全一样。凡是能够运行 MATLAB 核心的配置均能很好地运行 SIMULINK。

7.1.2　SIMULINK 的安装

SIMULINK 的安装与 MATLAB 的安装类似。关于它在 MS-Windows 下的安装方法,可以参考本书的第二章关于 MATLAB 的安装及其他有关的书籍。安装完后,会自动在 MATLAB 的目录结构中加入 SIMULINK 子目录,并自动修改 MATLAB 的启动文件 matlabrc.m。

7.1.3　SIMULINK 入门

为了让读者较快认识 SIMULINK，以一简单系统为例，从建模到仿真分步叙述如下：

(1) 在 MATLAB 指令窗运行指令 simulink，便打开图 7.1-1 所示的 SIMULINIK 模块库。

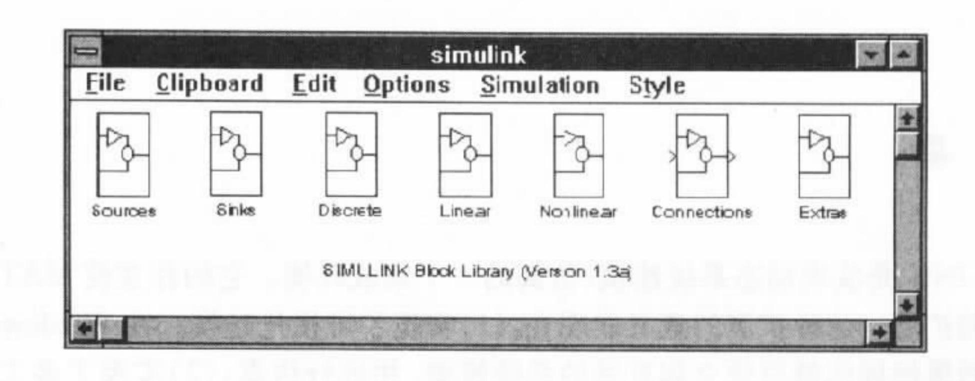


图 7.1-1　SIMULINK 标准模块库

(2) 选择【File】菜单中的【New】命令创建一个"Untitled"空白窗。

(3) 用鼠标双击信号源(Source)子库方框打开信号源模块库，并调整该信号源窗口和"Untitled"空白窗互不重叠，如图 7.1-2 所示。

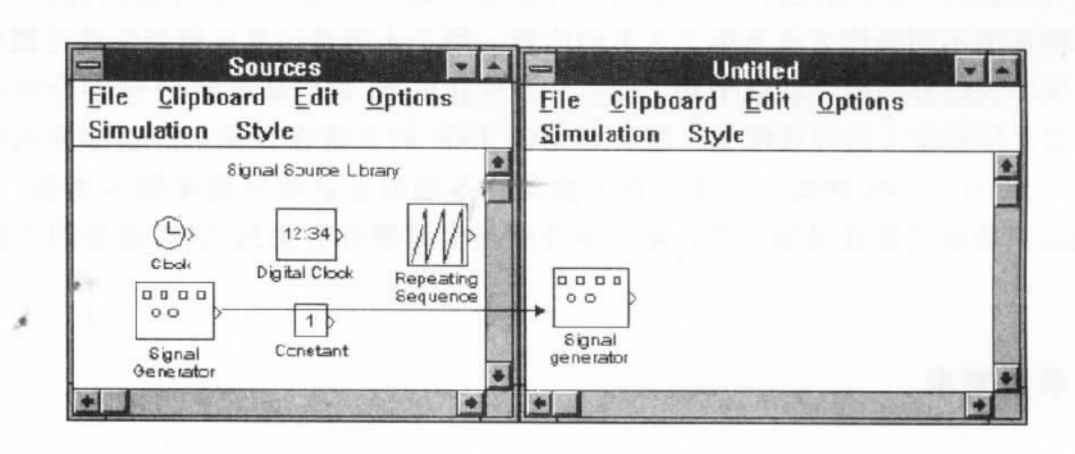


图 7.1-2　信号发生器模块的复制

(4) 将信号发生器(Signal Generator)模块从子库窗口拷贝到"Untitled"窗口。方法是：先把鼠标的光标移到信号发生器的方框图中，按住鼠标的左按钮，将它拖到新的窗口中，然后松开按钮，复制过程就完成了。该操作完成后的情况如图 7.1-2 所示。

(5) 模块内部参数的设置。被拷贝到新窗口中的模块包含着与原始模块一样的内部参数设置。如果想改变信号发生器的参数设置，用鼠标双击它的图标，就会弹出一个如图 7.1-3 所示的对话框，显示出对信号波形、幅值和频率的控制。

在图 7.1-3 中，亮的正弦波是当前选取的波形，频率(Frequency)是 1，幅值(Peak)为 1，其中【Frequency】、【Peak】和【Range】输入框中的数值表示当前的设置值，可以在其中输入数值重新设置它。另外还能用鼠标通过控制滑标来设置频率和幅值。在完成了参数的设置后，单击

【OK】按钮保存设置。

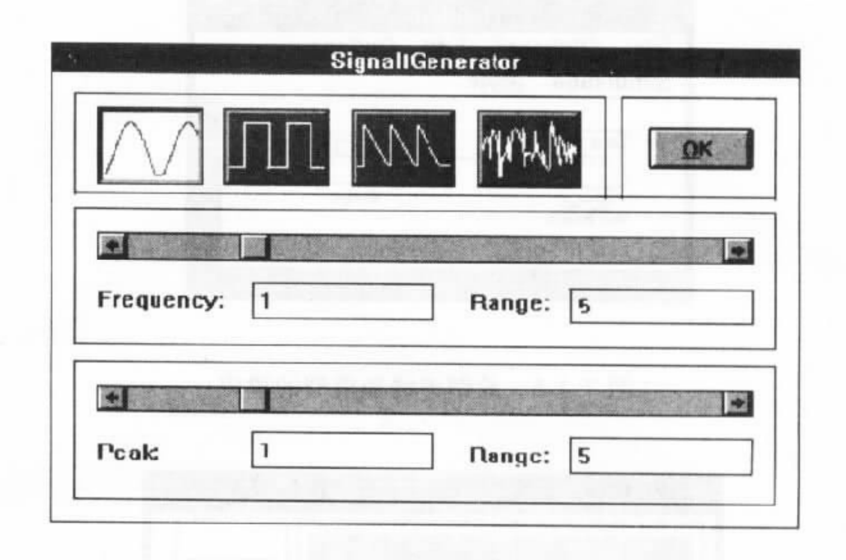


图 7.1-3 信号发生器的控制面板

用同样的方法打开信号显示模块库(Sinks),从中选取示波器(Scope),然后把它拷贝到新的系统窗口中,如图 7.1-4 所示。

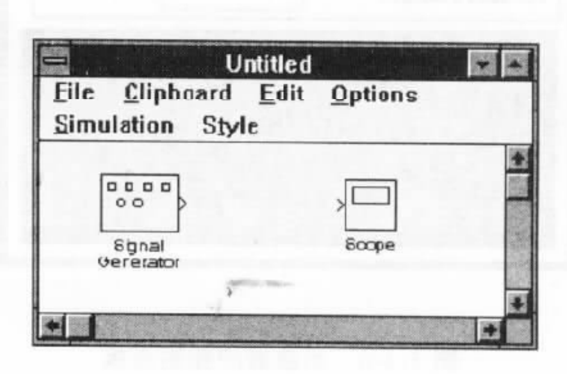


图 7.1-4 复制示波器

Signal Generator 和 Scope 外面的大于号(>)分别表示信号的输出和输入。为了连接这两个模块,使用鼠标的任一个按钮,点击输入或输出端口,看到光标变为(+)以后,拖动十字图符到另外一个端口,然后释放鼠标按钮,则带箭头的连线会表示信号的流向,如图 7.1-5 所示。至此,一个最简单的模型创建完成。

(6) 仿真操作一:信号发生器的参数设置。双击示波器图标后,出现一个虚拟示波器(图 7.1-6)。通过鼠标操作,使虚拟示波器与图 7.2-4 模型窗互不覆盖。选择【Simulation】菜单中的【Start】命令启动仿真过程。在虚拟示波器上将看到波形。

如果看不到希望的波形,应该先仔细选择示波器的时间轴范围(Horizontal Range)和波形幅值范围(Vertical Range)。

在本例中,因为信号是(圆)频率为 1 的单位幅值正弦波。所以,若想在示波器上看到一个周期以上的波形,就必须把示波器的时间范围设得大于 2π,而幅值范围应不小于 1;否则所显示的波形将被削截。

(7) 仿真操作二:算法参数的设置。

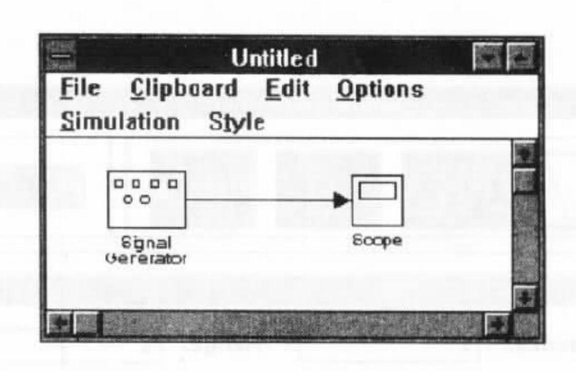


图 7.1-5 绘制连线和模型的建成

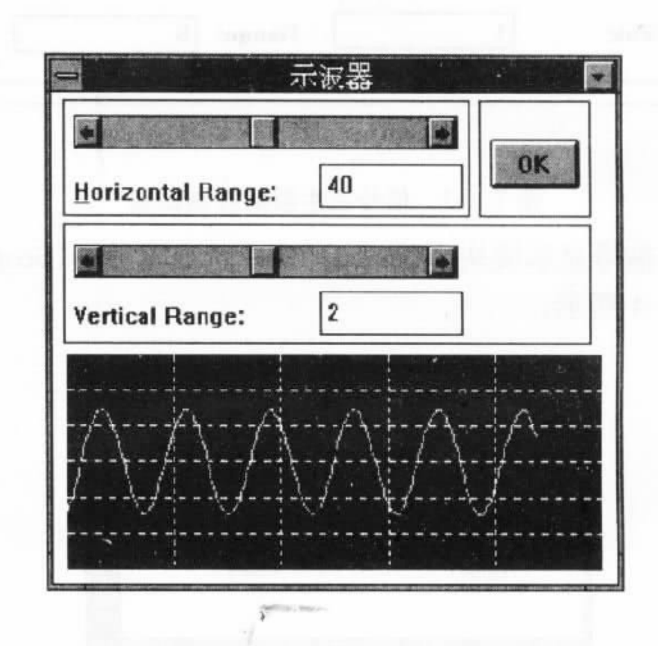


图 7.1-6 示波器的控制面板

在 SIMULINK 中，仿真中动态数据的计算都是由数值积分实现的。尽管本例从信号源到示波器没通过其他环节，但动态数据仍是经数值积分计算而得的。因此，适当选取算法参数是必须的。

算法参数的设置方法是：在图 7.1-5 所示的模型窗口中，在【Simulation】下拉菜单中选取【Parameters】选项。于是，出现一个名为“SIMULINK Control Panel”的对话框。把该框中的【Max Step Size】的默认值 10 改为 0.1（确保最大步长小于周期的 1/10）。

经过以上设置后，就可以在示波器上看到如图 7.1-6 所示的逼真波形。

关于算法参数设置将在下一节作更详细介绍。

7.1.4 界面与菜单

SIMULINK 工作窗是 MATLAB 窗口系列中的一种新类型。下面分别简要说明 SIMULINK 窗口中各菜单选项的含义，其中有些菜单的操作方法在后面的有关部分中将陆续详细介绍。

File 文件操作菜单

New	Ctrl + N	创建新的 SIMULINK 窗口
Open	Ctrl + O	打开已经存在的 SIMULINK 模型文件
Close	Ctrl + W	关闭当前的 SIMULINK 窗口
Save	Ctrl + S	保存当前的模型文件
Save As		将文件另外保存
Print	Ctrl + P	打印
Print Setup		打印设置
Exit MATLAB	Ctrl + Q	退出 MATLAB

Clipboard 剪贴板菜单

Copy	进行拷贝
Copy Options	拷贝选项

在拷贝选项运作时，会弹出对话框提供两种选项：(1)将方框图以 Windows 图元文件(文件的扩展名是 wmf)的形式剪贴；(2)将方框图以位图(文件的扩展名是 bmp)格式剪贴。

Edit 编辑菜单

Cut	Ctrl + X	剪切选定的内容
Copy	Ctrl + C	将当前选定的内容拷贝到剪贴板上
Paste	Ctrl + V	将剪贴板上的内容粘贴到当前光标所在位置
Clear		清除选定的内容
Select All		选择整个窗口

Options 建图操作菜单

Group	Ctrl + G	将选定的内容集合成组
Ungroup	Ctrl + U	把集合组还原为分立状态
Mask	Ctrl + M	封装一个新的 SIMULINK 模块
Unmask		打开一个封装好的模块
Flip Horizontal	Ctrl + F	将模块图水平旋转 180 度
Rotate	Ctrl + R	将模块图顺时针旋转 90 度
Reroute Line	Ctrl + L	重布连线，将模块间连线规范化

Simulation 仿真操作菜单

Start/Stop	Ctrl + T	仿真启动或停止
Restart		仿真重启动
Continue/Pause		仿真继续或暂停
Parameters		仿真参数的设置

Style 方框图样式菜单

Drop Shadows		加阴影效果
Orientation		模块输入输出口方向设置
Title		模块名称的设置
Font		字体的设置
Foreground Color		前景颜色
Background Color		背景颜色
Screen Color		屏幕颜色
Sample Time Color		给不同采样时间序列填加颜色
Wide Vector Lines		设置向量的线宽
Update Diagram	Ctrl+D	更新方框图

7.2 模型的构造

为了掌握 SIMULINK，应学习如何操作模块以及如何建立模型。对于 SIMULINK 的模块库（如图 7 1-1 所示）的内容这里不做介绍。本节着重对构造模型的过程作更为详细的介绍。

SIMULINK 完全采用方框图的“抓取”功能来构造动态系统。系统的创建过程就是绘制方框图的过程。在 SIMULINK 环境中方框图的绘制完全依赖于鼠标操作。鼠标的不同操作体现在光标的形状上。在缺省状态下鼠标是一个箭头，但在图形的操作中鼠标呈现不同的形状。

7.2.1 创建模型文件

建立 SIMULINK 模型通常有三种方式：

(1) 直接从 MATLAB 指令窗中选取【File】中【New】子菜单的【Model】命令，MATLAB 会打开一个新方框图窗。当然也可像第 7 1.3 节介绍的那样，在 MATLAB 指令窗下输入 simulink 命令，打开 SIMULINK 模块库窗口，然后再从它的【File】菜单中选取【New】命令，创建新方框图窗。

(2) 如果方框图模型已经存在，那么在 MATLAB 指令窗下直接键入模型文件名字，便会打开该模型的方框图窗口。用户可以对它进行编辑、修改和仿真。

(3) 由于 SIMULINK 模型是以 ASCII 码形式保存的 S-函数文件，所以还可以使用字处理软件对它进行创建、修改和编辑，这同建立一个普通 M 文件的过程一样，但要符合创建 S-函数的语法规则（关于 S-函数，以后介绍。）。

7.2.2 标准模块的选取

由图7 1-1 可见,SIMULINK 模块库按模块类型分为七个子库。用鼠标双击子库方框便可见到库中存放的各种标准模块。这些模块都可被复制到用户的模型窗中(见第7 1 3 节中的第四步)。

打开标准模块的另一个办法是:在 MATLAB 指令窗里键入 blocklıb 命令。该指令执行后,可以见到一组常用标准模块。

7.2.3 模块的移动、删除和拷贝

模块的移动

该操作非常简单:将光标置于待移动模块图标上,然后按住鼠标将模块拖到合适的地方即可。模块移动时,它与其他模块的连线也随之移动。

模块的选定

模块选定操作是许多其他操作(如删除、剪切、拷贝)的"前导性"操作。选定模块的方法有两种:

(1) 用鼠标单击待选模块,模块四个角处便出现小黑块,表示已经选定。

(2) 如果选择一组模块,可以按住鼠标按钮拉出一个矩形虚线框,将所有待选模块包在其中,然后松开按钮,则矩形里所有的模块同时被选中。

模块的删除、剪切和拷贝

对选定模块的删除、剪切和拷贝操作分述如下:

(1) 按【Delete】键,把选定模块删除。

(2) 选择【Edıt】菜单中【Cut】命令后,便将选定模块移到 Wındows 的剪贴板上,可供【Paste】命令重新粘贴。

(3) 运行【Edıt】菜单中【Copy】命令;然后将光标移到将粘贴的地方,按一下鼠标;看到选定的模块恢复原状,再运行【Edıt】菜单中的【Paste】命令,就会在选定的位置上复制出相应的模块。新复制的模块和原模块的名称也会自动编号,以资区别。

另一种更简单的复制操作是:先按下【Ctrl】键不放,然后用鼠标把待拷贝模块拖到希望位置后,松开鼠标左键,便完成拷贝工作。

7.2.4 模块的连接

关于模块连接,在本章一开始就已经涉及。值得提醒的是:连接模块时,要注意模块的输入输出口和各模块间的信号流向。在 SIMULINK 中,模块总是由输入口接受信号,从输出口发送信号。

【例 1】建立如图 7.2-1 所示的模型。

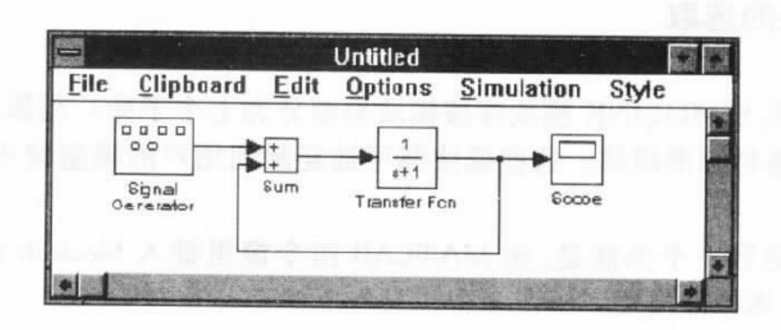


图 7.2-1　一个简单系统的构造

(1) 利用 SIMULINK 窗中【File】菜单的【New】选项开辟一个新的“Untitled”窗口。

(2) 分别从信号源库(Sourse)、线性模块库(Linear)、数据接受器库(Sinks)中,用鼠标把信号发生器(Signal Generator)、求和模块(Sum)和传递函数模块(Transfer Fcn)、示波器(Scope)拖取到“Untitled”窗口。各模块的位置如图 7.2-1 所示。

(3) 图 7.2-1 中模块间的连线有两类:一类是“直”线,它从某模块的输出口出发直指另一模块的输入口。这种连线的生成方法在第 7.1.3 节中已经讲过。另一类连线是“折”线。这类“折”线的生成方法是:把鼠标光标移到传递函数模块和示波器间连线的中点附近,按下鼠标右键,光标由箭头变为十字,往下拉动鼠标到适当位置后放开右键,屏幕上就出现一条由中点引出的箭头线,再从此箭头开始用鼠标水平向左画线到适当位置,再松开鼠标键,照此操作,直到整个“折”线绘完。

7.2.5　模块属性的改变

模块的属性可以分为两种:模块的标题和模块的内部参数。为适合自己的需要,那些被用户所复制的标准模块的标题和参数常需作必要的修改。下面就介绍修改方法。

标题的修改

模块标题是指标识模块图标的字符串。对用户所建模型窗中模块标题进行修改的具体方法如下:

(1) 用鼠标单击标题,使之增亮反显。

(2) 输入新的标题(中西文均可)。

(3) 然后用鼠标点击窗口中的任一地方,修改工作结束。

模块内部参数的修改

双击待修改参数模块的图标,打开参数设置对话框,然后通过改变对话框适当栏目中的数据便可。下面通过具体算例来说明具体操作。

【例 1】把图 7.2-1 方框图模型修改成图 7.2-2 所示的模型。注意到本系统是以文件 sim.m 保存的。

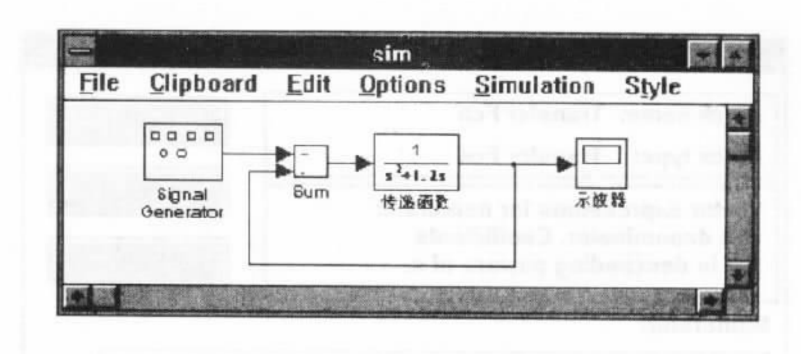


图 7.2-2 系统 sim.m

(1) 对标题的修改

用鼠标点亮传递函数块标题“Transfer Fcn ”，输入汉字字符“传递函数”。再点亮示波器模块标题“Scope”，输入汉字字符“示波器”，再在窗口空白处按点鼠标左键，标题修改结束。

(2) 对求和块输入极性的修改

用鼠标双击求和块(Sum)，就会弹出对话框。

把【List of signs】栏中的缺省极性改成如图 7.2-3 所示形式，再点【OK】键，原求和模块图标便自动改成图 7.2-2 所示形式。

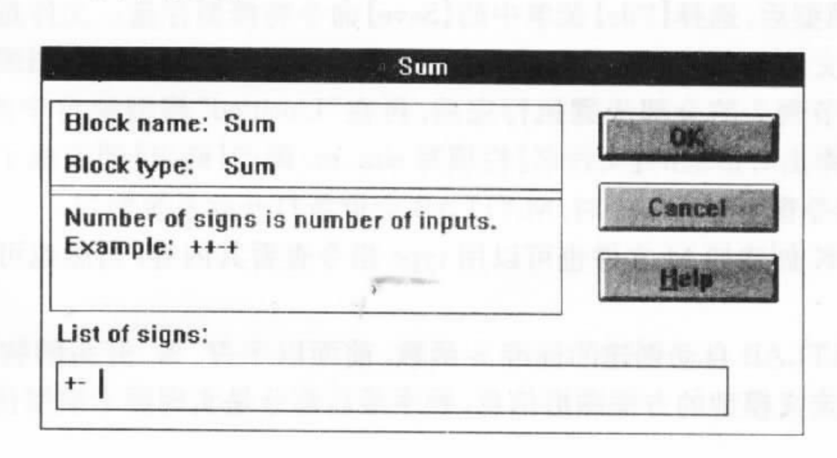


图 7.2-3 求和模块对话框

(3) 对传递函数块参数的修改

用鼠标双击传递函数模块图标，弹出如图 7.2-4 所示对话框。

【Denominator】栏表达的是传递函数的分母多项式系数向量。把该栏中的原缺省值改成图 7.2-4 所示的行向量，再点【OK】键，原传递函数模块图标中的函数表达式就自动变成图 7.2-2中的形式。

值得指出的两点是：(1)在参数设置时，任何 MATLAB 工作内存中已有的变量、合法表达式、MATLAB 语句等都可以填写在设置栏中；(2)模块图标的大小是可以用鼠标操作调整的。因此，假如传递函数表达式太长，原方框容纳不下，可以用鼠标把它拉到适当大小，使整个方框图标美观易读。

图 7.2-4 传递函数模块对话框

7.2.6 模型文件的保存

构造好一个模型后,选择【File】菜单中的【Save】命令将模型存盘。文件是以 ASCII 码形式存储的 M 文件。文件包含了该模型的所有信息(数学模型内涵和外部框图表现)。

例如,当上小节例 2 的全部步骤执行完后,再在"Untitled"模型视窗中的【File】菜单中选【Save】子项,并在弹出对话框的【文件名】栏填写 sim.m,再点【确定】便完成了文件的保存。以后在 MATLAB 指令窗下运行 sim 时,MATLAB 会重新打开此系统窗口。

由 SIMULINK 创建的 M 文件也可以用 type 指令查看其内容,当然也可以用字处理器进行修改和编辑。

此文件是 MATLAB 自动创建的标准 S-函数,前面以字符"%"开头的物理行是函数的注释行,紧接着程序定义模型的方框图形信息,程序最后部分是实现数学模型仿真运算的指令。

7.3 数值分析

仿真涉及常微分方程组的数值积分。由于动态系统行为的多样性,目前还没有一种算法能够保证所有模型的数值仿真结果总是准确、可靠的。为此,SIMULINK 有一组不同的数值积分算法供用户选择。因此,为得到准确仿真结果,用户必须针对不同模型,仔细选择算法及仿真参数。

7.3.1 菜单操作方式下仿真算法和参数的选择

所谓菜单操作方式的仿真是指,仿真是在模型的 SIMULINK 视窗下进行。这种仿真方式最直观。在这种仿真方式下,无论是对框图模型本身还是对数值算法及参数的选择都可以很方便地修改和操纵。

模型参数和框图的实时操作

模型不仅在仿真前允许编辑和修改，而且在仿真过程中也允许作一定程度的修改。在菜单操作方式下，允许对模型和框图进行如下实时操作：

(1) 被仿真模块的参数允许实时修改，限制条件是：该参数的变化不改变模型的结构（包括模型几何结构、输入输出维数及状态空间维数）。

(2) 离散模块的采样时间允许实时修改。

(3) 允许用游离示波器（Floating Scope）实时观察任何一点或几点的动态波形（所谓游离示波器是指在模型视窗里与系统模型没有任何可见连接线的示波器）。

(4)在一个系统仿真的同时，允许打开另一个系统。

算法和算法参数的操作

在仿真模型的 SIMULINK 窗口菜单条中，【Simulation】下拉菜单中的【Parameters】选项是用来设置仿真算法和参数的。运行【Parameters】命令后，SIMULINK 会弹出仿真控制面板（SIMULINK Control Panel），如图 7.3-1 所示。

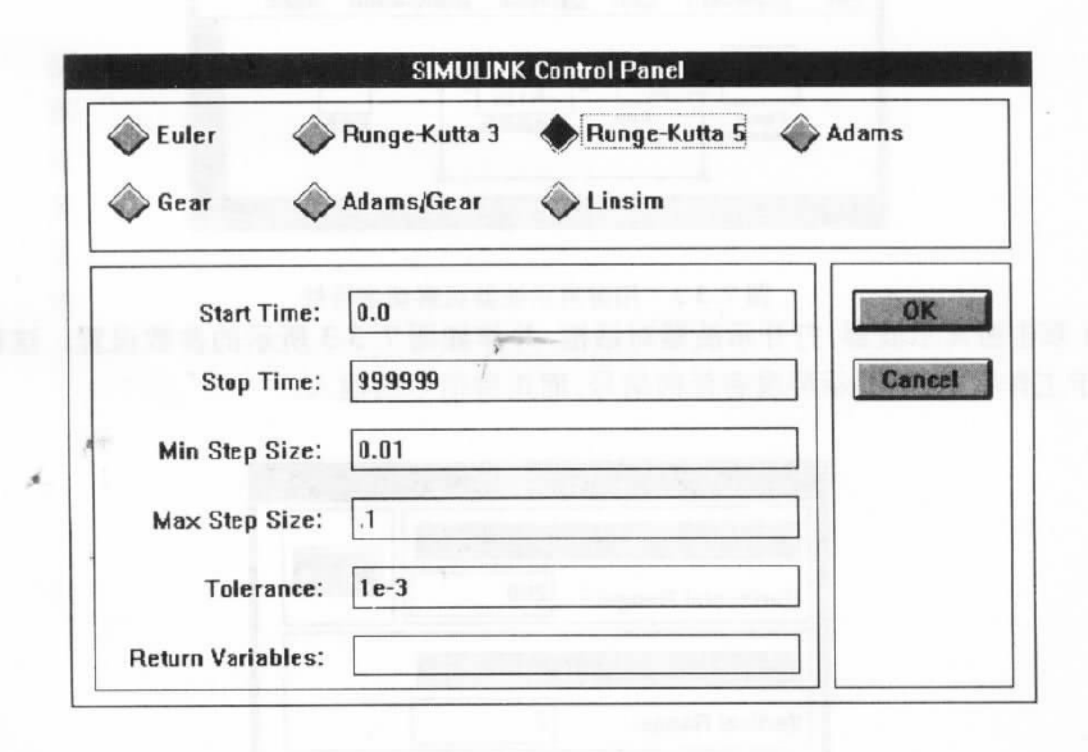


图 7.3-1 算法和参数的选择面板

关于图 7.3-1 选择面板作如下说明：

(1) 面板上方有七个积分算法单选按钮。缺省算法是五阶龙格-库塔法（Runge-Kutta 5）。关于积分算法在后面将做详细介绍。

(2) 各栏含义如下：

Start Time	仿真的起始时间（不允许实时修改）
Stop Time	仿真的停止时间（仿真过程中允许实时修改）

Min Step Size	积分的最小步长(仿真过程中允许实时修改)
Max Step Size	积分的最大步长(仿真过程中允许实时修改)
Tolerance	容差(仿真过程中允许实时修改)
Return Variables	返回 MATLAB 工作内存的变量(不允许实时修改)

可以通过在栏目中填写适当的数字、已存在变量的名字、表达式,去替代默认值。

(3) 缺省情况下,返回变量栏是空白。在这种情况下进行仿真时,只能在仿真过程中用示波器现场观察动态特性。一旦仿真结束,在 MATLAB 的工作内存中不留下任何记录。返回变量的正确形式是三元输出参数组 [t, x, y]。各参数分别代表时间、系统状态和输出。

【例 1】用游离示波器观察。

(1) 把图 7.2-2 所示 sim.m 系统改变成图 7.3-2 所示的系统,即示波器呈游离状态。

操作方法:用鼠标单击图 7.2-2 框图中示波器输入口的黑箭头,于是到这示波器的连线显示出小黑方块的句柄,再按键盘上的【Delete】键,便得到图 7.3-2。

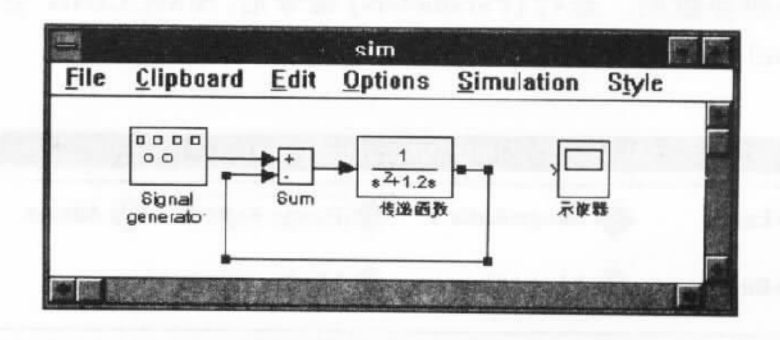


图 7.3-2 用游离示波器观察动态特性

(2) 双击游离示波器,打开示波器对话框,并作如图 7.3-3 所示的参数设置。这样示波器已经处于工作状态,其显示屏没有任何信号,而正待信号的输入。

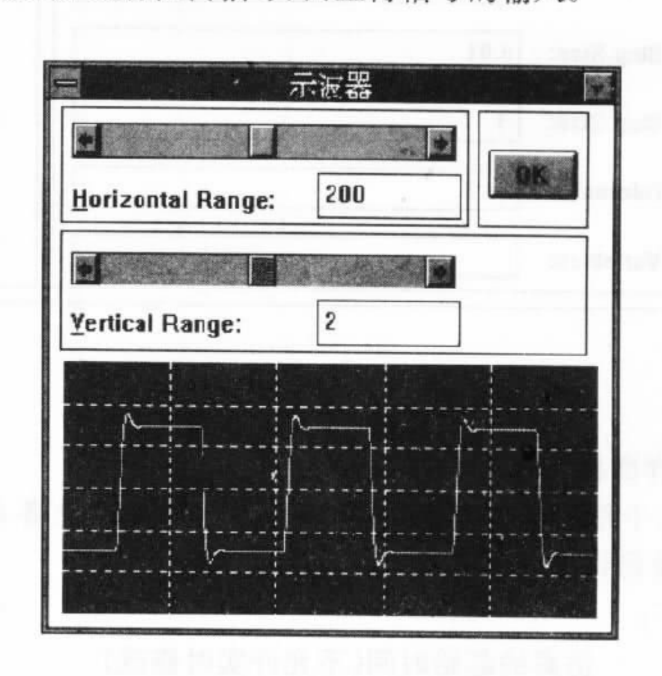


图 7.3-3 用游离示波器观察波形

(3) 用鼠标单击模型框图反馈线，这意味着把游离示波器的输入线接到该反馈线上。

(4) 双击打开信号发生器的对话框，选择方波信号，设置频率为 0 1，然后关闭对话框。

(5) 选择【Sımulınk】菜单中的【Start】启动仿真，在示波器中就可以看到图 7. 3-3 所示的波形。读者可以试着实时地调整模型参数、算法及算法参数，观察波形的变化。若要停止系统的仿真，选择【Sımulınk】菜单中的【Stop】命令。

7.3.2　仿真的 MATLAB 指令操作方式

数值积分指令的调用格式

任何在 SIMULINK 视窗中建立的方框图模型都可以在 MATLAB 指令窗中被调用、仿真。在 MATLAB 的内部函数中六种数值积分算法：lınsım, rk23, rk45, adams, gear, euler 等。这六种指令的调用格式相同，下面以 lınsım 为例加以说明，其调用格式为：

lınsım('model', tf, xı, optıons, ut, p1, …, p10)

[t, x, y] = lınsım('model', tf, xı, optıons, ut, p1, …, p10)

说明：

(1) 在这两种调用格式中，除第一、第二输入参数外，其余输入参数都可以缺省。

(2) 输入参数 model 是模型的(M 或 MEX)文件名，为字符串变量。模型可由方框图生成，也可直接写成。这是在任何一种调用格式中必不可少的。

(3) 输入参数 tf 是仿真时区。当 tf 为标量时，默认仿真时区是[0 , tf]；当 tf 为二元行向量时，仿真的时间区间是[tf(1) , tf(2)]。

(4) 第三个输入参数 xı 是系统初始状态。在仿真的菜单操作方式下，不能设置初始状态。

(5) 第四个输入参数 optıons 是仿真算法参数设置向量。它的每个分量的含义如下：

optıons(1)　相对误差(缺省值为 0 001)

optıons(2)　最小步长 (缺省设置是 [tf(2) - tf(1)]/2000)

optıons(3)　最大步长 (缺省设置为 [tf(2) - tf(1)]/50)

optıons(4)　算法切换开关 (缺省设置设置为 0，即算法不切换)

optıons(5)　积分过程参数显示开关(缺省为 0，表示不显示。)

optıons(6)　没有输出参数时的绘图开关(缺省为 1，表示绘制图形)

(6) 输入参数 ut 是被仿真系统的外部输入变量。它可以是字符串或数值表。如字符串 'one(2, 1) * sın(3 * t)' 表示二元输入列向量。输入数值表的格式是第一列为时间序列，其余每列代表在该时间序列上的各输入分量。

(7) 从第六个输入参数起的各参数是传送模型参数用的。对于由框图构造的模型，一般都不进行模型参数传送。

(8) 输出参数的含义为：

t　取积分值的时间点序列向量。

x　系统的状态序列矩阵。

y　系统的输出序列矩阵，每列表示一个输出的时间序列。

(9) 没有输出参数的调用格式。在 optıons(5) 的缺省设置下，在运行结束后将给出图形曲

线,或者是系统输出曲线,或者是系统状态轨线(当模型框图上无输出口时)。

MATLAB 指令操作仿真的优点

与用菜单仿真相比,从 MATLAB 指令窗中进行系统的仿真有以下几个好处:

(1) 可以重新设置模块的初始值。

(2) 如果在调用函数时不指定输出变量,则会自动绘制系统的状态轨迹,或者在有输出模块时绘制系统的输出。

(3) 可以指定比较复杂的外部输入函数 ut。

(4) 用字处理器编写的 M 文件和 MEX 文件(当然包括由 SIMULINK 框图建造的模型文件)都可以仿真。

(5) 仿真中,可以动态地改变模块参数。这为系统仿真带来了极大的灵活性。

指令操作仿真前模型框图输入模块的选择

在前面所构作的框图模型中,系统的输入信号是由信号发生器产生的。这种信号源主要用在仿真的菜单操作方式中。在仿真的指令操作方式中,阶跃函数信号(Step Input)是最常用的一种信号源。它很适用于瞬态特性的研究。

阶跃输入模块的阶跃时刻、"跃前"函数值、"跃后"函数值都可设置。在此特别提醒注意:缺省的阶跃时刻是 1,而不是 0。

指令操作仿真前模型框图输出模块的选择

无论是图 7.2-2 还是图 7.3-2,观察输出用的都是示波器。在菜单操作方式的仿真中,这种示波器用起来十分方便。但是,在 MATLAB 指令方式下,调用这种框图模型文件进行仿真时,就会发现问题:无法从仿真中得到输出波形和数据。其原因在于:示波器只是现场观察模块,它不具备保存数据的能力。

在指令操作方式的仿真中,常用到 SIMULINK 模块库中的两个模块:输出接口(Output);数据暂存模块(To Workspace)。下面分别用算例来说明它们的使用方法。

【例 1】对图 7.3-4 所示框图模型进行仿真的指令操作方式。

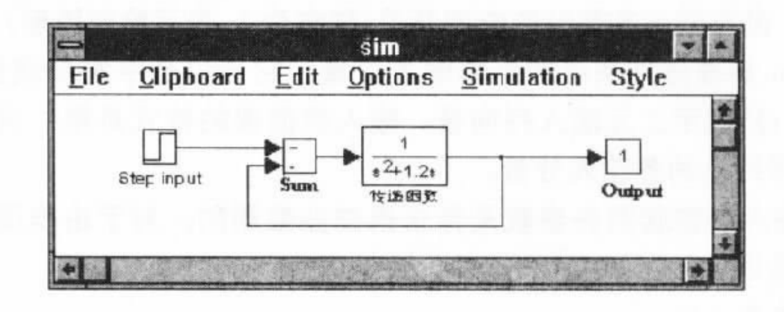


图 7.3-4　带输出接口的模型框图

(1) 图 7.3-4 中的阶跃输入模块可从信号源模块库复制而得;输出接口模块则从连接模块库拷贝。

(2) 在图 7.3-4 所示的模型框图视窗里,选择【File】下拉菜单中的【Save】,存储框图模型在

D: \ mywork \ sim. m 文件中(D: \ mywork shi 是由用户自己定的目录名)。

(3) 在 MATLAB 指令窗(或在 Notebook)中,运行以下指令就可得到如图 7.3-5 所示的输出响应。

```
cd d: \ mywork
[t,x,y]=linsim('sim',20);
plot(t,y)
```

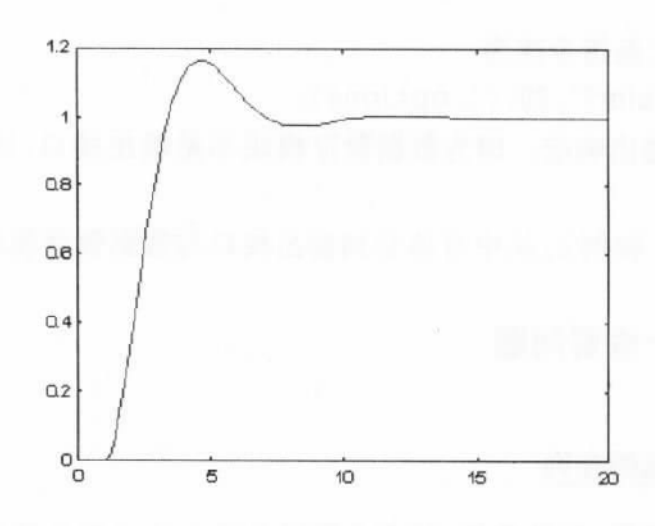


图 7.3-5 系统的输出响应曲线

说明:

(1) 由于阶跃输入模块工作在缺省值状态,所以响应曲线在 t=1 以前是零。

(2) 只有使用输出接口模块,才能使上述第二条指令运行后获得输出数据 y。

【例 2】利用数据暂存模块,进行指令操作方式的仿真。

(1) 构作图 7.3-6 所示的框图模型。数据暂存模块在信号源模块库(Sources)中。

(2) 双击接在输出端的数据暂存模块,打开其对话框,在【Virable Name】栏中填写 y,关闭此对话框,便可看到该模块图标名称自动变成图 7.3-6 所示的样子。用同样方法把数据暂存模块 1 定名为 t。模型框图中的时钟信号模块是必须的,它由信号源模块库复制而得。

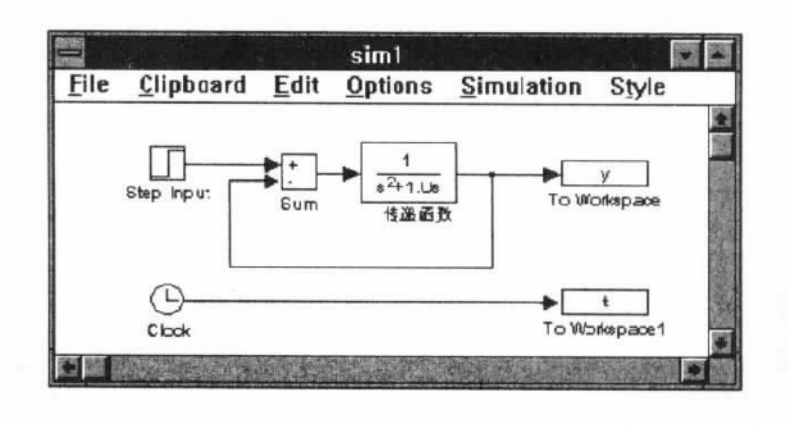


图 7.3-6 利用数据暂存模块构作的模型

(3) 把所做框图模型存放为文件 sim1 m。

(4) 在 MATLAB 指令窗(或在 Notebook)中,运行以下指令就可得到与图 7 3-5 一样的响应曲线。

```
options(6)=2;          % 该设置目的在于抑制下面指令的图形显示
linsim('sim1',20,[],options)
plot(t,y)
```

说明:

(1) 假如把上述第二条指令改为

```
[t,x,y]=linsim('sim1',20,[],options);
```

那么将不可能获得输出响应。因为数据暂存模块不是输出接口,所以这条指令运行所得的 y 输出将是“空”[ ]。

(2) 请仔细比较例 1 和例 2,从中可体会到输出接口与数据暂存模块的不同。

7.3.3 仿真中的几个重要问题

模型变量和初始状态的查询

模型中设置的各种参数:初始条件、状态个数以及输入输出的个数等,都很容易查询而得。在此不作一般讲述,而仅通过算例作若干说明。

【例 1】图 7 3-4 所示系统的信息查询。

```
[sizes,x0,xstr]=sim          % sim 是系统 sim m 的文件名
sizes =
     2
     0
     1
     0
     0
     0
     1
x0 =
     0
     0
xstr =
/sim/传递函数
/sim/传递函数
```

说明:

(1) 上述返回参数 sizes(1)=2 表示连续状态变量数为 2;sizes(3)=1 表示输出口数为 1。

(2) 返回参数 x0 表示该系统为零初始状态。

（3）xstr 表示两个状态变量都属传递函数模块所有。

初始条件的设置

在菜单方式下，总是采用零初始状态进行仿真。在指令操作方式下，仿真系统的初始条件是可以设置的。若数值积分指令中不出现第三输入参数（即初始状态参数）项或在第三参数项位置上是空阵[ ]，那么仿真时将采用零初始状态。

初始状态的设置很简单，只要在数值积分指令的第三参数位置上填写维数适当的已知向量便可。

【例2】使用图 7 3-4 所示系统进行初始状态不同设置的仿真。

（1）为习惯需要，把图 7 3-4 中的阶跃输入的阶跃时刻设置在 0。

（2）运行以下指令，可得图 7 3-7 所示的相轨迹图。

```
[t,x1,y1]=linsim('sim',20);
[t,x2,y2]=linsim('sim',20,[0.5,0]);
plot(x1(:,1),x1(:,2),'r:',x2(:,1),x2(:,2),'b-');
legend('零初始状态','非零初始状态')
```

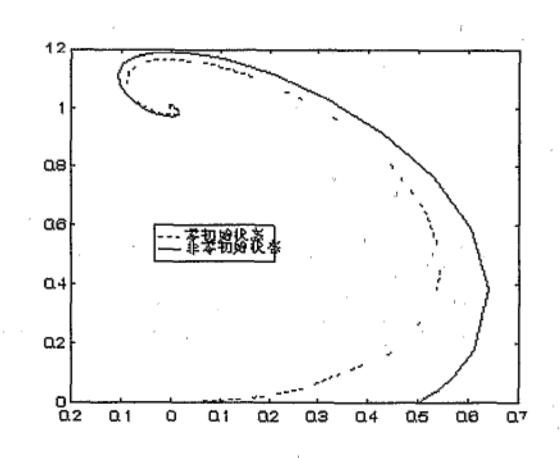


图 7 3-7 不同初始条件对相轨迹的影响

说明：第一条语句采用系统默认的全零初始条件；第二句采用自定初始条件（0 5，0）。它们的作用可以从相轨迹图上清楚地看到。

代数循环

在构造系统时应注意避免代数循环的发生，所谓代数循环就是几个直通模块构成了一个反馈环节。当发生这种情况后，仿真速度将大大变慢甚至无法进行下去。下面是几个常见的直通模块：

（1）增益模块（Gain）；

（2）大多数非线性环节，如查表模块（Look Up Table）、限幅模块（Limiter）等；

（3）分子分母同阶的传递函数；

(4) D 矩阵不为零的状态方程(State-Space)模块。

数据的插补

SIMULINK 积分算法的计算步长是自适应的，也就是说在积分过程中步长按精度的要求随时变化。比如在轨迹平坦的地方步长取得较大，而变化剧烈的地方步长则取得较小以满足积分精度。因此，对同一个问题，算法的不同、同一算法参数的不同，将产生出点数和取值位置都不同的计算数据。这就给不同仿真结果的比较带来困难。

此外，当计算数据较少，所画曲线不太光滑时，也要用到数据插补技术。

在 MATLAB 中有许多数据插补函数(如 table1，interp2，spline)可供选用。

【例 3】利用例 2 修改过的图 7 3-4 模型，进行算法精度控制的比较。

```
[t,x,y]=linsim('sim',10,[],[1e-8,1e-6,1]);
[t1,x1,y1]=linsim('sim',10,[],[0.1,1e-5,1]);
yint=table1([t,y],t1);
vector_error=norm([yint-y1])
per_point_error=sum(abs(yint-y1))/length(t1)
vector_error =
     0.0133
per_point_error =
     0.0011
```

说明：

(1) 被仿真系统的文件名是 sim m，积分时间为 10s，初始条件采用系统默认值。在前两句积分指令中分别定义了不同的算法参数：积分相对误差、最小步长和最大步长。

(2) 第三句是使用 MATLAB 的一维线性插补函数 table1，由高精度计算结果按低精度数据取值点位置进行插补，从而在相同的数据点上进行两组数据的比较。

六种数值仿真算法的特点

为对付各种不同的模型，SIMULINK 提供了表 7.3-1 中的六种求微分方程数值解的算法以供选择。

表 7.3-1　SIMULINK 仿真算法

仿真算法	算　法　说　明
linsim	专用于线性系统的仿真算法。如果系统是由线性模块(如传递函数、状态空间、求和以及增益模块等)组成的，或者仅包含少量的非线性环节，那么使用 linsim 是合适的。特别是对于一些病态(stiff)的线性系统(如有很快的动态环节)，linsim 要优于其他方法。
rk23	三阶龙格-库塔法。当系统为高度非线性或不连续时，使用龙格-库塔法是最好的。而且该方法适用于连续和离散的混合系统，但是当系统同时有快速和慢速动态环节时，龙格-库塔法的效果不太好，这时应使用 linsim 或 gear 法。
rk45	五阶龙格-库塔法。

表 7.3-1(续)

仿真算法	算 法 说 明
gear	专为解算病态系统而设计的基尔预估-矫正法。但对于非病态系统,特别是当系统奇异或受到快变输入扰动时,该法不如其他方法有效。
adams	阿达姆斯预估-矫正法。该方法适用于光滑、非线性,时间常数变化范围不大的系统。
euler	欧拉法。精度差,仅用来验证结果。

步长控制

在数值积分指令中,如

[t, x, y] = linsim('model', tf, x0, [tol, minstep, maxstep])

其中,第四个输入参数 options 的前三项 [tol, minstep, maxstep] 定义了积分的相对误差、最小步长和最大步长。如何设置这三个参数,对仿真结果的精度有较大的影响。下面是设置这三个参数时,需注意的几个方面:

(1) tol 控制积分步长的相对误差,通常在 0 1 和 1e-6 之间取值。tol 越小,积分的步数就越多,精度也越高。但是过小的 tol(如 1e-10),未必会给出很高的精度,原因是.那时的计算截断误差将显著增加。

(2) 最小步长是仿真开始时所用的步长。在仿真开始后,算法会把算得的局部估计误差与设定的 tol 相比较,在满足 tol 的前提下,自动拉大步长,提高计算效率。因此,产生一个数据输出点所取的步长通常不会小于最小步长。唯一的例外情况是离散采样周期小于最小步长。

通常,最小步长都取得很小,如 1e-6。但要注意:如果最小步长设置过小,则当系统含间断点时,会产生大量的数据点。这会过于消耗计算机资源。

对 adams 和 gear 法来说,最小步长不影响解的精度,只影响输出数据点的个数。因此,在这两种方法中,应设最小步长等于最大步长,其值应使输出数据足够用以绘制光滑曲线和便于分析。

(3) 有时仿真精度虽已达到要求,但并不产生光滑的轨迹。此时,有必要减小最大步长以产生光滑的图形结果。例如对于有逐段线性输入的线性系统,linsim 可以取任意大的步长而不损失精度;又如 rk45 也可能会在满足精度的情况下取较大的步长。

(4) 尽管 euler、linsim、rk23 和 rk45 是变步长积分方法,但当最大步长等于最小步长时,仿真将采用相应的定步长积分法。

(5) 积分函数在计算每一个输出点时,除了 linsim 和 euler 法只需调用一次模型外,其他的积分方法都需要几次调用模型来计算系统状态的微分。因此,积分步长被分割,产生实际步长小于最小步长的现象。例如 rk45 每输出一个点按(0, 1/4, 3/8, 12/13, 1, 1/2)比率计算六步;rk23 按(0, 1/2, 1)比率计算三步。而 adams 和 gear 法因为是采用预估-矫正法,所以在获得每一计算点时所需的实际步点是不定的。

7.3.4 离散系统的仿真

SIMULINK 具有仿真多频采样离散系统和离散-连续混合系统的能力。

离散模块

SIMULINK 的每个离散模块都有一个内置输入采样器和输出零阶保持器。当离散模块和连续模块混用时,离散模块在采样间隔内的输出保持不变,而对离散模块的输入仅在采样的瞬间才被更新。

采样周期

离散模块采样周期的设置是,通过双击离散模块后所打开的对话框(图 7.3-8)实现的。

Discrete Transfer Fcn

Block name: Dis. Transfer Fcn

Block type: Discrete Transfer Fcn

Vector expressions for numerator and denominator. Coefficients are in decending powers of z.

OK

Cancel

Help

Numerator:

[1]

Denominator:

[1 2]

Sample time:

[Ts, offset]

图 7.3-8 离散传递函数模块的参数设置对话框

在【Sample time】栏里,若仅填写一个标量参数,那么它就是采样周期。若在此栏中填写二元向量,那么该向量的第一个元素指定采样间隔 Ts,第二个元素设置偏移时间 offset。实际采样时刻为:

t = n * Ts + offset

在此,n 为整数,offset 是绝对值小于采样时间 Ts 的实数。若要求模型必须在某时刻更新,就必须借助 offset 的设置来实现。

纯离散系统

纯离散系统可使用任何一种积分算法进行仿真,而不会影响输出结果。若只要采样瞬间的输出数据,那么应把最小步长设置得比最大的采样间隔大。

离散-连续混合系统

离散-连续混合系统由离散模块和连续模块混合组成。仿真时,可以采用前面介绍过的任何一种方法。但是对大多数混合系统来讲,龙格-库塔变步长积分法(rk23 和 rk45)要优于其他方法,而不推荐使用 adams 和 gear 积分法。

多频采样系统

多频采样系统包含有不同采样速率的离散模块。在 SIMULINK 中,多频采样系统,乃至多频采样-连续混合系统的建模和仿真都可以进行。

【例 1】图 7.3-9 所示双速率采样系统 dsys.m 的仿真。

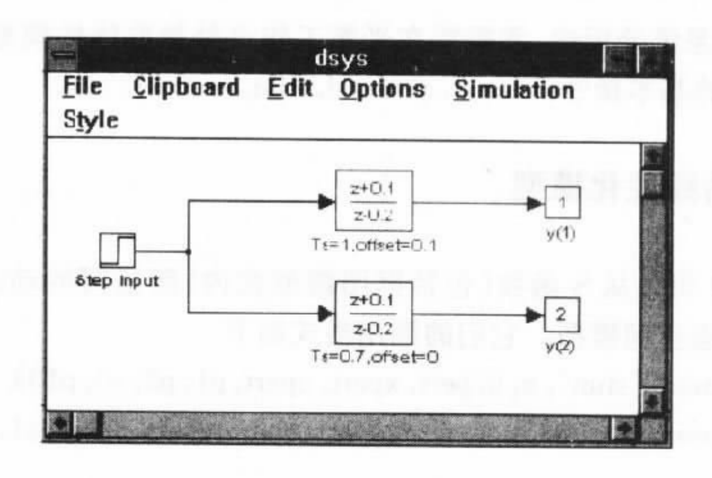


图 7.3-9 多采样速率的离散系统

(1) 建立图 7.3-9 所示的框图模型。设两个离散传递函数模块的采样时间和偏移时间分别为[1 , 0.1] 和 [0.7 , 0]。用【Style】菜单中的【Sample Time Color】命令可以用颜色显示出两个模块采样时间的不同。

(2) 运行以下指令即可,如图 7.3-10 所示。

```
[t,x,y]=euler('dsys',3);
stairs(t,y)
legend('y(1)','y(2)')
```

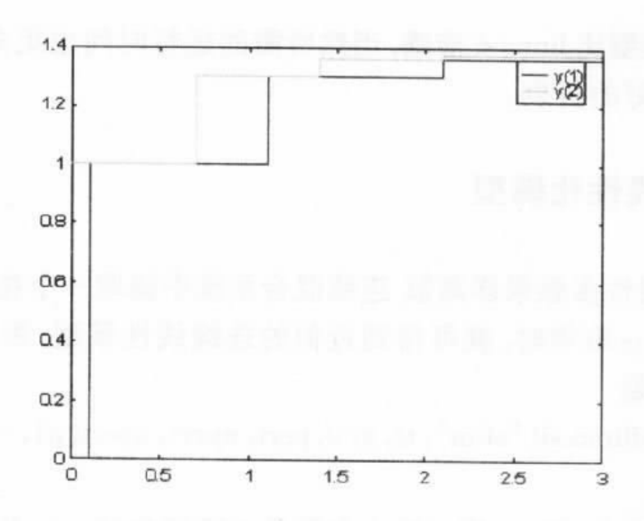


图 7.3-10 不同采样速率模块的输出结果

说明:

(1) 本例中阶跃输入函数的阶跃时刻设置为 0 。

(2) 仿真所得系统输出 y 矩阵的第一列为模块 1 的输出，第二列为模块 2 的输出。

(3) 由于为黑白输出，故无法表现原图的彩色特性以及图例的标注。

7.4　仿真系统的线性化模型

在一般的非线性系统分析中，常需要在平衡工作点处提取线性模型。为此，SIMULINK 提供了三个十分有用的基本指令：linmod, dlinmod, trim。

7.4.1　连续系统的线性化模型

linmod 和 linmod2 可以从 S-函数（包括框图模型在内）所描写的动态系统中提取一个用 [A, B, C, D]表达的状态空间模型。它们的调用格式如下

[A, B, C, D] = linmod('sfun', x, u, pert, xpert, upert, p1, p2, …, p10)

[A, B, C, D] = linmod2('sfun', x, u, pert, apert, bpert, cpert, dpert, p1, p2, …, p10)

说明：

(1) 第一输入参数'sfun'是必不可少的，sfun.m 是被仿真系统的 S-函数文件名。

(2) 第二、第三输入参数 x 和 u 分别设定系统的状态和输入工作点。缺省值为全零设置。

(3) 最后的十个参数 p1～p10 是向 sfun 系统传送参数用的，它们可以缺省。当然采用全局变量也可以实现 sfun 系统与外界的参数交换。

(4) 第四个输入参数 pert 是全局扰动因子，它仅在紧跟其后的分类扰动因子缺省时产生作用。在 linmod 中，缺省值为 1e-5；在 linmod2 中，缺省值是 1e-8。

(5) 在 linmod 指令中的第五、第六输入参数 xpert 和 upert 分别是状态扰动因子和输入扰动因子，允许缺省。在 linmod2 中，第五到第八输入参数分别是状态方程四元矩阵组的扰动因子，允许缺省。

(6) linmod2 所得模型比 linmod 准确，当然所需的运行时间也更多。linmod2 会在圆整误差和截断误差之间取最好的折衷。

7.4.2　离散系统的线性化模型

dlinmod 能够从非线性多频采样离散-连续混合系统中提取一个在任何给定的采样周期 ts 下的近似线性模型。当 ts 取零时，就可得到近似的连续线性模型；否则，得到离散线性模型。该指令的一般调用格式是

[Ad, Bd, Cd, Dd] = dlinmod('sfun', ts, x, u, pert, xpert, upert, p1, …, p10)

说明：

(1) 在任何调用方式中，第一、第二输入参数是必不可少的。ts 是指定采样周期的。

(2) 在原系统稳定的前提下，若 ts 是原系统所有采样周期的整数倍，则由 dlimod 所得线性模型在 ts 采样点上与原系统有相同的频率响应和时间响应。即便在上述条件不满足的情况下，该指令仍可能给出有效的线性化模型。

(3) Ad 的特征根是原系统稳定性的一个很好指示。当 ts=0 时,若 Ad 的所有特征根在左半平面,则系统稳定;当 ts>0 时,Ad 特征根在单位圆内,则指示系统稳定。

(4) 如果原系统不稳定或 ts 不是原系统采样周期的整数倍,所得 Ad、Bd 有可能是复数阵。即便如此,Ad 的特征根仍能为稳定性判断提供相当的信息。

(5) dlınmod 十分有用。它可以把系统从一种采样周期模型变换到另一种采样周期下的模型,可以把离散模型变成连续模型,也可以把连续模型变成离散模型。

7.4.3 关于模型线性化的几点说明

(1) 若被线性化系统本身是线性的,那么线性化过程不存在截断误差,扰动可取任意大小,工作点也不影响所得结果。一般说来,扰动取得较大,有利于减小圆整误差。

(2) 在线性化过程中,模型的状态次序不变。

(3) 在 SIMULINK 框图上,系统的输入输出要依次编号。

(4) SIMULINK 框图上各模块与状态变量的对应关系可用第 7 3 3 节例 1 所讲的方法查询。

(5) 所获得的线性模型,可以用控制工具包、鲁棒控制包及其他工具包作进一步的处理。

7.4.4 平衡点的确定

在给定输入、输出及状态条件下,系统平衡工作点的求取由 trım 指令实现。它的一种典型调用格式是

[x, u, y, dx] = trım('sfun', x0, u0, y0, ıx, ıu, ıy)

在此,'sfun'是被研究的系统文件名。ıx, ıu, ıy 都是整数向量,它们的元素指示初始猜测向量 x0, u0, y0 中哪些分量将固定不变。

除非问题本身的最小值唯一,否则不能保证所求得的平衡点是最佳值。因此,若想寻找全局最佳平衡点,那么多尝试几组初始猜测值是很有必要的。

当系统有不连续状态时,trım 一般不适用,而 trım4 也许能给出较好的结果。

【例 1】求图 7 4-1 所示系统的线性模型、Bode 图及平衡工作点。

(1) 在 SIMULINK 中建立如图 7 4-1 所示的模型。注意:作图中,反馈模块输入输出口方向的改变可以用【Optıons】下拉菜单中的【Rotate】实现;非线性饱和模块(Saturatıon)使用的是缺省设置(即上下限为 +0 5 和 -0 5)。

(2) 运行以下指令,可得到线性模型的[A , B , C , D],传递函数及 Bode 图,见图7 4-2。

```
[A,B,C,D]=linmod('lin')
[num,den]=ss2tf(A,B,C,D);
printsys(num,den,'s')
bode(A,B,C,D)
A =
    -1.0000          0         1.0000
```

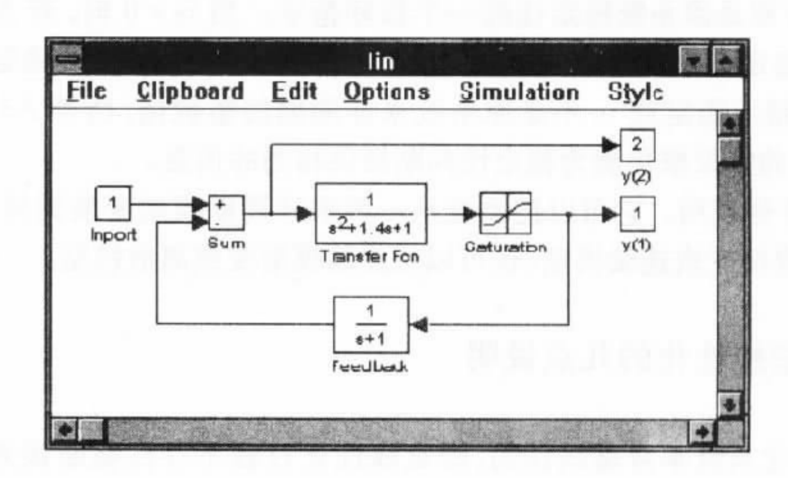


图 7.4-1　单输入双输出系统

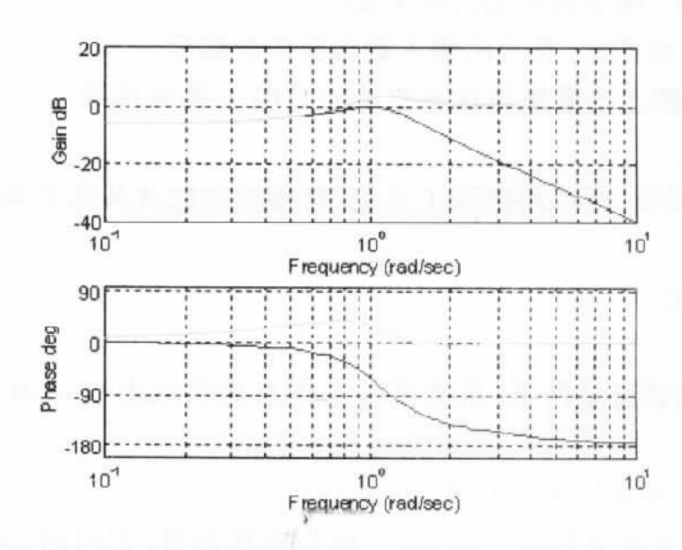


图 7.4-2　系统线性化模型的 Bode 图

```
   -1.0000    -1.4000    -1.0000
         0     1.0000          0
B =
     0
     1
     0
C =
     0     0     1
    -1     0     0
D =
     0
     1

num(1)/den =
```

```
  -1.776e-015 s^2 + 1 s + 1
  -------------------------
  s^3 + 2.4 s^2 + 2.4 s + 2

num(2)/den =
  s^3 + 2.4 s^2 + 2.4 s + 1
  -------------------------
  s^3 + 2.4 s^2 + 2.4 s + 2
```

(3)运行以下指令求平衡点。

```
x=[0;0;0];
u=0;
y=[1;1];
ix=[];              % 不固定任何状态
iu=[];              % 不固定输入
iy=[1;2];           % 固定输出 y(1)和输出 y(2)
[x,u,y,dx]=trim('lin',x,u,y,ix,iu,iy)
x =
    0.5000
    0.0000
    0.7727
u =
    1.2727
y =
    0.5000
    0.7727
dx =
  1.0e-024 *
         0
         0
   -0.1213
```

说明:

(1) 在算得的 y(1)对输入的传递函数中,分子 s^2 项的系数在 1e-15 数量级,这是由计算截断误差产生的,它实际为零。

(2) 最后一项输出 dx 的数量级为 1e-24,这也是截断误差产生的,实际应为零。在不同的机器上,得出的数值会不一样,但数量级不变。

7.5 S-函数及其应用

S-函数是 SIMULINK 运作的核心。在 SIMULINK 环境中,若要充分发挥该软件的建模和仿真能力,那么不管用什么方式建立模型文件,都必须理解和掌握 S-函数。

S-函数有三种表现形式:框图形式;M 文件形式;MEX 文件(C 或 FORTRAN 子程序)。

在使用中，这三种方式各有优缺点。框图表示比较直观，容易构造，运行速度比较快；MATLAB 文件编写灵活，适用面宽，运行较慢；MEX 文件运行速度最快。因此，使用何种方式应视具体情况而定。在解决较复杂问题时，常常需要不同方法交叉使用。

本节将介绍：S-函数的本质；S-函数的三种创建方式；S-函数不同形式间的转化；用户自定义模块的“封装”；子系统的构作；参数设置和传送。

7.5.1　什么是 S-函数

在 SIMULINK 视窗中的系统方框图一经建立，SIMULINK 就会利用该框图中的信息生成一个 S-函数（即系统函数），S-函数是 SIMULINK 如何运作的核心所在。

每个框图都有一个与之同名的 S-函数，而该 S-函数正是 SIMULINK 在仿真和分析中交互作用的载体。简单地说，S-函数代表了 SIMULINK 模型。

至关重要的一点是要意识到：S-函数从本质上讲是具有特殊调用格式的 MATLAB 函数，它表征系统动态特性。如此处理的优点是：

(1) 可以通过 M 文件或方框图创建所需的线性、非线性模型；

(2) 可以创建模块库中没有的新模块；

(3) 可以编写所需的分析、仿真程序。

尽管 SIMULINK 图形交互方式创建框图模型的 S-函数对用户是隐视的，但 S-函数对用户毕竟是透明的，是可以触及的。假如直接用 S-函数去处理问题，那将在复杂系统的分析和仿真中表现出很强的功能。

尽管 SIMULINK 会自动为方框图创建 S-函数，但用户也可以用标准的 M 文件、C 或 FORTRAN 程序编写 MEX 文件去直接定义 S-函数。直接定义 S-函数的基本思想是：必须向 SIMULINK 提供仿真变量的明确定义；变量随时间的变化是如何由当前状态变量和输入决定的；系统的输出是什么。

对于很多用微分方程描述的动态系统来说，用 M 文件或 MEX 文件建立 S-函数比方框图建模更为方便。S-函数不管用什么方式创建，一旦建立，它既可以在框图中使用，也可以在指令中调用。

7.5.2　S-函数的工作方式

引入 S-函数的目的是为了使 SIMULINK 有能力构作一般的仿真方框图，去处理如下各种系统的仿真：连续系统、离散系统、离散-连续混合系统、多频采样系统、嵌套系统等。这些功能是由 flag 变量实现的：通过将 flag 设置成不同的值，可以使 S-函数体现系统的各个不同侧面。下面是 flag 取不同数值时，S-函数所表现的系统行为。

通常 S-函数的调用格式是：

sys = sfunction(t, x, u, flag)

其中，sfunction 是用户定义的系统，t 是当前时刻，x 是当前状态值，u 是当前系统输入值，而变量 flag 的值控制返回变量 sys 的信息。

表 7.4-1 flag 各选项的作用

flag	S-函 数 的 表 现
0	返回系统变量和初始条件的维数
1	返回系统的状态导数 dx/dt
2	离散状态 x(n+1)
3	返回系统的输出向量 y
4	更新下一个离散状态的时间间隔

如果令 flag=0,调用 S-函数,即指令为

[sizes, x0]=sfunction([],[],[],0)

那么返回参数 x0 表示状态向量的初始值,而返回参数 sizes 各分量的含义如下:

sizes(1) 连续状态变量数

sizes(2) 离散状态变量数

sizes(3) 输出变量数

sizes(4) 输入变量数

sizes(5) 系统中不连续根的数量

sizes(6) 系统有无代数循环的标志(有,则置 1)

在运用 S-函数进行仿真运算时,必须清楚地知道系统不同时刻所需要的信息。例如开始进行仿真时,应先知道:系统有多少状态变量,其中哪些是连续变量,哪些是离散变量,这些变量的初始条件等信息;而这些信息的获取,可由在 S-函数中设置 flag=0 获取。进而,若系统是严格连续的(不含离散环节),那么在每一仿真时步中需要知道的是:给定时刻的系统状态导数(令 flag=1 可得)和系统输出(令 flag=3 可得)。若系统是严格离散的(不含连续环节),那么只需令 flag=2 获得下一个离散状态,令 flag=3 获取离散系统的输出。

7.5.3 创建 S-函数

在前面了解 S-函数的基础上,本节通过一个实例来详细说明如何使用三种不同的方法(方框图、M 文件和 MEX 文件)创建 S-函数,并比较其异同。

用方框图创建 S-函数

考虑著名的范德坡方程(Van de Pol Equation):

$$\frac{d^2x}{dt} + (x^2 - 1)\frac{dx}{dt} + x = 0$$

写成状态方程形式为:

$$\begin{cases} \dfrac{dx_1}{dt} = x_1(1 - x_2^2) - x_2 \\ \dfrac{dx_2}{dt} = x_1 \end{cases}$$

于是可构造 SIMULINK 方框图如图 7 5-1 所示。

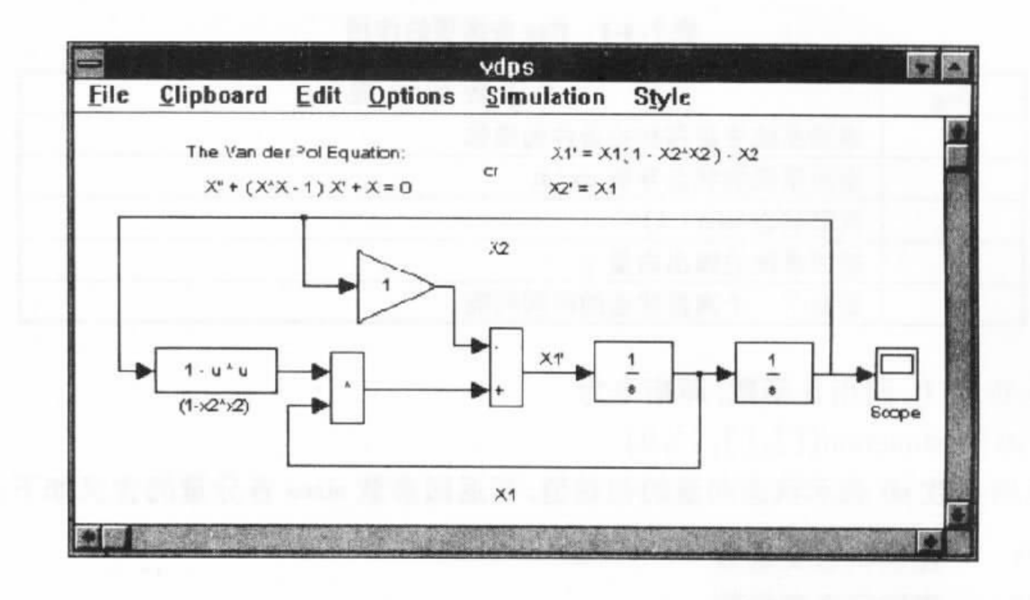


图 7.5-1 范德坡方程的方框图

说明:

(1) 在方框图形成的同时,与之相应的 S-函数(即 M 文件)也随之产生;当生成的框图被保存后,相应的 S-函数就记录在磁盘上。这个 M 文件包含了该方框图所有的图形及数学关系信息。然而,当在框图视窗中进行仿真时,MATLAB 并非去解释运行该 M 文件,而是运行保存于 SIMULINK 内存中的 S-函数映象文件。

(2) 当方框图以 vdps.m 保存后,就可以在 MATLAB 中访问该系统,并通过设置 flag 的值得到系统的动态信息。详见以下各例。

【例 1】设置 flag=0,查询 vdps 系统的维数和初始条件。

```
[sizes,x0]=vdps([],[],[],0)
sizes =
      2
      0
      0
      0
      0
      0
x0 =
      0.2500
      0.2500
```

说明:返回变量 sizes 的各分量表明:该系统有 2 连续状态,没有离散状态,没有输入和输出,状态是连续的,没有代数循环。变量 x0 给出两个状态的初始值。

【例 2】设置 flag=1,获取状态对时间导数。

```
flag=1;
```

```
u=[];
t=0;
x=[0 25;0.25];
dx=vdps(t,x,u,flag)
dx =
    0.2500
   -0.0156]
```

用 M 文件创建 S-函数

当系统以微分方程表征时，可以用 M 文件更简练地编写描述系统动态特性的 S-函数。用 M 文件编写范德坡方程的程序如下：

vdpm.m

```
function [sys, x0] = vdpm(t, x, u, flag)
if abs(flag) = =1,
  sys=zeros(size(x));
  sys(1)=x(1)  * (1-x(2) ^2)-x(2);
  sys(2)=x(1);
elseif flag= =0
  sys=[2;0;0;0;0;0];
  x0=[0 25;0 25];
else
  sys=[];
end
```

用 C 语言(MEX－文件)创建 S-函数

用 C 创建(MEX 文件)S-函数时，可编写两个函数分别定义系统的初始条件和状态微分。下面是 C 语言写的源程序，使用 Borland C++ 4 0 编译器，并利用 MATLAB\BIN 目录下的批处理文件 cmex bat 将其编译成 vdpmex dll 文件。

vdpmex.c

```
#define NSTATES 2
void init-conditions(double * x0)
{
      x0[0]=0 25;
      x0[1]=0 25;
}
void derivatives(double t, double * x, double * u, double * dx)
{
      dx[0] = x[0] * (1-x[1] * x[1])-x[1];
      dx[1] = x[0];
```

```
}
#include <simulink h>
```

说明：

(1) 编译命令 cmex vdpmex c，产生 MS-Windows 下的动态链接库文件 vdpmex dll。这是 Windows 下的可以执行映象，可以在 MATLAB 下调用。例如，若用指令[sys, x0] = vdpmex ([],[],[],0) 去查询系统维数和初始状态，可得到与例 1 相同的结果。

(2) 在用 C 语言编写程序时必须包含“头文件 ”simulink h，它提供了 MEX 文件到 MATLAB 的标准界面。宏定义 # define NSTATES 2 说明了系统是一个二阶系统。

(3) 在用 C 语言编写 S-函数时，子函数的名字是固定的，不能随便改动。在 matlab \ simulink 目录中有很多用 C 语言编写的范例，可以用做参考。

(4) 对 MATLAB 4 2c 版本，要求的 C 语言编译器必须为 Borland C+ + 4 0 以上版本。如想使用其他编译器，那么可以参考 MATLAB 的 readme txt 文件。

S-函数的仿真运算

上述三种方式所创建的 S-函数都可以用于仿真，但运行速度不同。MEX 文件形式的 S-函数运行速度最快，框图的仿真速度次之，M 文件形式的 S-函数运行速度最慢。

【例 3】用两种不同的 S-函数解算范德坡方程，并比较运行速度。

(1) 运行以下指令，用两种不同的 S-函数解算范德坡方程，并画出状态变化曲线，见图 7 5-2。

```
[ts,xs]=rk45('vdps',10);
[tm,xm]=rk45('vdpm',10),
plot(ts,xs,'o',tm,xm,'-'),
```

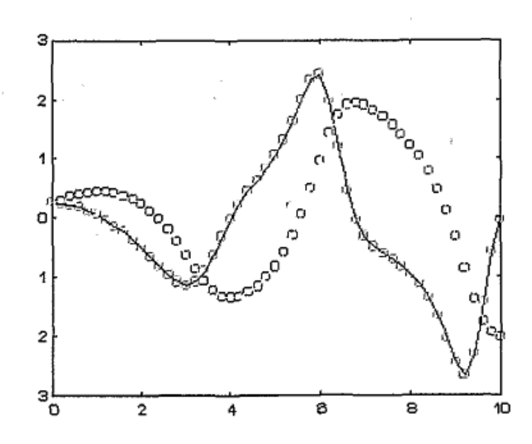


图 7 5-2　由不同 S-函数获得相同的范德坡方程解算结果

(2) 运行以下指令，比较两种 S-函数的仿真时间。仿真在 486/DX-80 的 PC 计算机上进行。

```
t0=clock; [ts,xs]=rk45('vdps',100), etime(clock,t0)
```

```
t0=clock; [tm,xm]=rk45('vdpm',100); etime(clock,t0)
ans =
    0.2200
ans =
    5.1600
```

说明:从结果可以看出,由方框图构成的S-函数的仿真速度比M文件快得多。

7.5.4 S-函数文件转化为框图模块

任何一种方式创建的S-函数文件,在经过通用S-函数模块(S-function block)处理后,将转变为用户自创建的SIMULINK模块,并且利用这种新模块仿真不会降低效率。

下面以具体算例来说明S-函数文件是如何转化为框图模块的。

【例1】用M文件编写一个限幅积分器的S-函数,并借助通用S-函数模块调用此M文件。限幅积分器的数学模型为:

$$x = \begin{cases} o & (x \leqslant lb, u < 0) \text{ 或}(x \geqslant ub, u > 0) \\ u & \text{其他} \end{cases}$$

其中,x为状态,u是输入,lb和ub分别表示积分的上限和下限。

(1) 据数学模型,用M文件编写如下S-函数。

lim_int.m

```
function [sys, x0]=lim-int(t, x, u, flag, lb, ub, xi)
if abs(flag)==1
    if (x<=lb & u<0) | (x>=ub & u>0)
        sys=0;
    else
        sys=u;
    end
elseif flag==3
    sys=x;
elseif flag==0
    sys=[1;0;1;1;0;0];
    x0=xi;
else
    sys=[];
end
```

(2) 直接运行M文件,观察运算结果。

```
x0=[];              % 积分函数的初始条件
options=[];         % 使用积分参数的缺省设置
lb=-0.5;            % 积分值下限
```

```
ub=0.5;                 % 积分值上限
xi=0;                   % S-函数限幅积分器的初始条件
rk23('lim-int',10,x0,options,'cos',lb,ub,xi)
```

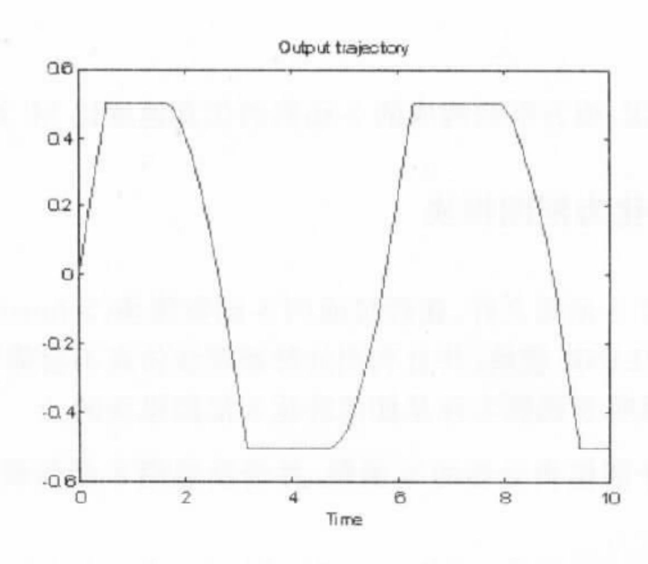


图 7.5-3　余弦输入下限幅积分的输出

(3) 创建以下 SIMULINK 视窗(图 7.5-4)。框图中,Sine-Wave 正弦信号模块复制于 SIMULINK 信号源模块库,并且利用其缺省设置。Scope 示波器的水平方向设置为 10,垂直方向设置为 1。S-Function 模块复制于非线性模块库(Nonlinear)。

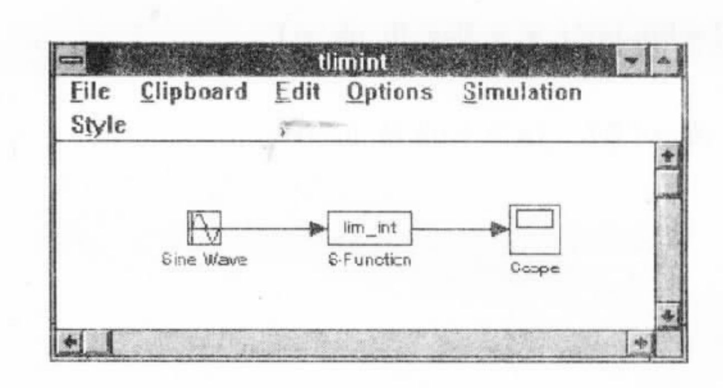


图 7.5-4　借助 S-函数模块调用 M-文件

(4) 用鼠标双击新复制的 S-函数模块图标,打开如图 7.5-5 所示的对话框。在对话框的最下方两栏中填写如图 7.5-5 所示的内容。

【Subsystem function name】栏中应填入 S-函数文件名 lim-int;在【Function parameters】栏中填入要传送的三个变量 lb、ub、xi(假若,S-函数文件除了 t、u、x 和 flag 输入变量以外没有其他的输入变量,那么最后一栏不用填写。),然后按【OK】。这样处理后,原通用 S-函数模块图标符就自动改写为 lim-int,该模块也就成了有限积分模块。

(5) 在图 7.5-4 所示的视窗里就可以进行仿真实验。双击示波器图标,打开示波器;然后选择【Simulation】下拉菜单中的【Start】,就可以从示波器上看到经限幅积分后的截顶正弦波。

说明:图 7.5-4 系统只是一个借助 S-函数通用模块调用 M 文件的示例。该例所述的方法

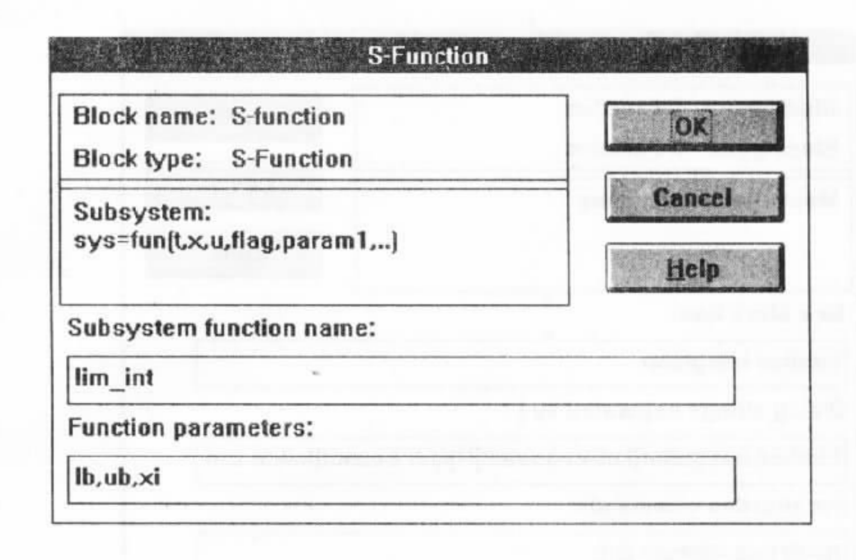


图 7.5-5 通用 S-函数模块的对话框

适用于其他系统框图。经过 S-函数对话框定义后，M 文件就可以像其他标准模块一样直接在框图视窗中参与运作。

7.5.5 改变模块的属性

SIMULINK 提供的封装(masking)功能允许您改变模块(包括子系统和 S-函数模块)的属性。通过将 M 文件和 MEX 文件编写的 S-函数封装起来，可以完全地把任何动态系统放进 SIMULINK 中。

图 7.5-4 已经显示用 S-函数模块将 M 文件置于 SIMULINK 中，但在进行仿真运算时还要与 MATLAB 工作间进行数据交换，即必须先在 MATLAB 工作间中定义三个变量 ub、lb 和 xi 的值，这无疑是不方便的。下面用例子说明如何把上节的限幅积分函数 limint 封装成一个真正的 SIMULINK 模块。

【例 1】把限幅积分函数 limint 封装成一个真正的 SIMULINK 模块。

(1) 模块名称的改写

在图 7.5-4 所示系统中，用鼠标单击 S-函数模块的标题，并将其改为"Limited Integrator"。

(2) 制作新的对话框和图标符

点击选中图 7.5-4 所示框图视窗中的 S-函数模块，再选择【Options】菜单中的【Mask】命令，于是出现图 7.5-6 所示的对话框。

(a)【Block Type】栏的填写

该项定义新模块的类型(用户可以根据自己需要分类)。本例模块类型填为 limint。

(b)【Dialog String】栏的填写

该项定义新模块的参数设置对话框。新的对话框中每个选项都用符号"|"分隔。第一项为对话框的标题，其余的为编辑选项的提示信息。最多可以有六个对话框提示信息。

在本例中，要求新对话框的形式是：标题为" Limited integrator"；第一选项提示为"Lower

S-Function

Block name: S-Function
Block type: S-Function
Mask Block Definitions

OK
Cancel
Help

New block type:
Limited Integrator
Dialog strings separated by | :
Limited integrator|Lower bound:|Upper bound:|Initial con
Initialization commands:
lb=@1;ub=@2;xi=@3;
Drawing commands:
plot([0,1,2,3],[0,0,1,1]);F(s)=1/s
Help string:
ited integrator with an\nexpression of the form:\n\nif (x<

图 7.5-6　Masking 对话框

limit:"；第二选项提示为"Upper limit:"；第三选项提示为"Initial condition"。为此，在【Dialog String】栏中应填写如下字符串：

Limited integrator|Lower limit:|Upper limit:|Initial condition

如果待"封装"模块在鼠标双击时，不是打开参数设置对话框，而是直接运行某 MATLAB 命令，那么在【Dialog String】栏中应使用 eval 指令。例如，若在【Dialog String】栏中输入：eval('plot(1:10)')，那么该"封装"模块被双击时，将会绘制一条直线。

(c)【Initialization Commands】栏的填写

该项为初始化命令，是该"封装"模块在图示和仿真时所要运行的命令。用作"封装"模块新对话框中参数的 MATLAB 表达式运算值可以通过@1、@2……传递。在本例中模块需要三个传入参数：lb(积分下限幅)、ub(积分上限幅)和 xi(积分器初始条件)，那么在【Initialization Commands】栏应填写

lb=@1;ub=@2;xi=@3;

(d)【Drawing Commands】栏的填写

该项用来绘制新"封装"模块的图标符。在【Drawing Commands】栏中允许填写：(A)任何表达式作为图标符；(B)指令 plot(x,y,x1,y1)，以便绘制指定形状的曲线作为图标符。

在此的 plot 命令与 MATLAB 中的一般 plot 类似，但有更严格的限制：不能指定线型和颜色；x,y 必须成对(或以复向量)传入；曲线坐标只能自动刻度。

在本例中，【Drawing Commands】栏中填写

plot([0,1,2,3],[0,0,1,1]);F(s)=1/s

(e)【Help String】栏的填写

该栏填写内容将成为，新"封装"模块对话框中与【Help】按钮对应的弹出帮助信息。

在本例中，填写以下内容：

Implements a limited intehrator with an \ nexpression of the form：

\ n \ nif (x< = lb and u<0) or (x> = ub and u>0) \ nxdot = 0 \ nelse \ nxdot = u \ nend

注意：在上面填写内容中，每遇到\ n 就回车换行。

(f)全部填写后的 Masking 对话框如图 7.5-6 所示。选择【OK】，于是原模块就被封装结束。封装后的新模块图标见图 7.5-7。

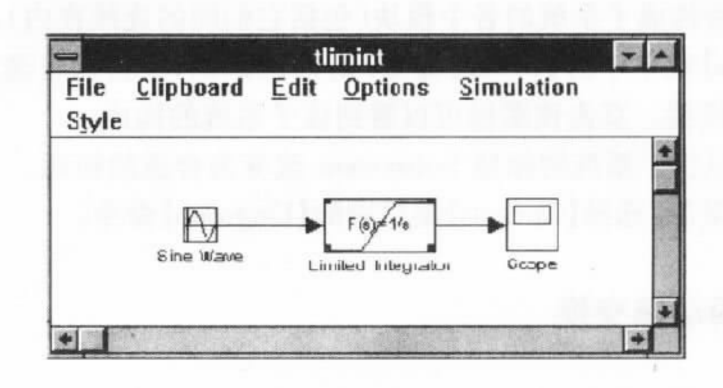


图 7.5-7　封装后的新模块图标

(g) 双击图 7.5-7 所示的限幅积分模块图标，会显示出新制作成的对话框(图 7.5-8)。

Limited Integrator (Mask)

Block name: Limited Integrator

Block type: Limited Integrator (Mask)

Limited integrator

OK

Cancel

Help

Lower bound:

-0.5

Upper bound:

0.5

Initial condition:

0

图 7.5-8　封装后新模块的对话框

(h) 按图 7.5-8 填写上、下限和初始条件后，便可运行。

像其他标准模块一样，这个封装后的模块具有独特的图标和方便易用的对话框。通过"封装"技术，用户可以建立自己的 SIMULINK 模块、模块库。

如果以后要更改模块的属性，可重新调用【Mask】，然后进行修改。若要解除"封装"，可使用【Unmask】命令实现。

7.5.6 创建子系统

SIMULINK 还提供创建子系统(Subsystem)的功能。子系统的建立有利于管理大型系统。当一个动态模型包含许多环节时,最好是把系统按功能分块,建立子系统。事实上,在 SIMULINK 的模块库里有许多模块(如 PID)本身就是由多个更基本模块构成的子系统。建立子系统的方法非常简单,主要有以下几个步骤:

(1) 用鼠标选定待构成子系统的各个模块(包括它们间的连线在内)。

(2) 运行【Options】菜单中的【Group】命令,则 SIMULINK 自动将选定范围内的模块及连接用 Subsystem 图标代替。双击该图标可以看到该子系统的构成。

(3) 如有必要可以将子系统的标题 Subsystem 改变为合适的标题。

如果想把子系统复原,选择【Options】菜单中的【Ungroup】命令。

7.5.7 模块参数的动态交换

本节进一步介绍框图模型与其他文件、工作内存之间的数据交换。

在 MATLAB 工作内存中定义变量

框图模块运作所需的参数和初始变量取自模块对话框。而模块对话框中填写的 MATLAB 变量以及表达式又来自 MATLAB 工作内存。不管仿真以何种方式进行,总可以在 MATLAB 工作内存中为 SIMULINK 模块预定义参数和初始变量,也可以在指令窗或命令文件中交互地进行变量的数值传递。

【例 1】在 MATLAB 内存中预定义框图模块参数。考虑一个单输入,双输出的状态方程:

$$x = Ax + Bu$$
$$y = Cx + Du$$

其中,矩阵 A、B、C、D 和初试条件向量 x0 分别为:

$$A = \begin{bmatrix} -0.3 & 0 & 0 \\ 2.9 & -0.62 & -2.3 \\ 0 & 2.3 & 0 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 1 & 0 \\ 1 & -3 & 1 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, x0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}。$$

(1) 构造如图 7.5-9 所示的框图系统,将其保存为 s-sys.m 文件。打开状态空间模块(State-space)的对话框,将 A、B、C、D 分别填入参数输栏,在初始条件栏中填 x0。

(2) 运行以下指令得图 7.5-10 所示曲线:

```
A=[-0.3,0,0;2.9,-0.62,-2.3;0,2.3,0],
B=[1;0;0];C=[1,1,0;1,-3,1];D=[0;1],
x0=[1;1;1];
[t,x,y]=rk23('s-sys',10);
plot(t,y(:,1),':b',t,y(:,2),'-r');legend('y1','y2')
```

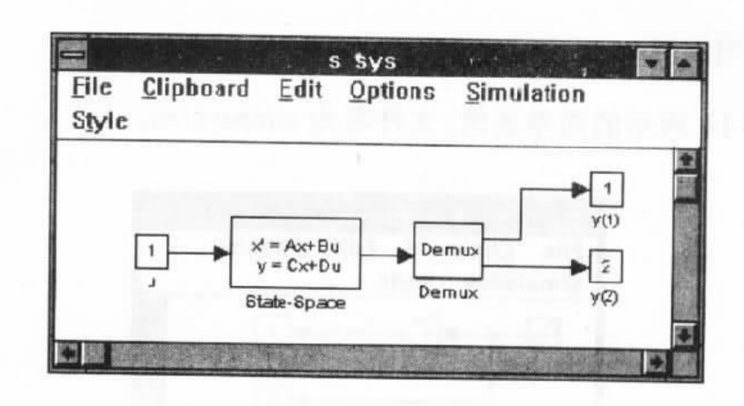


图 7.5-9 仿真系统的框图模型

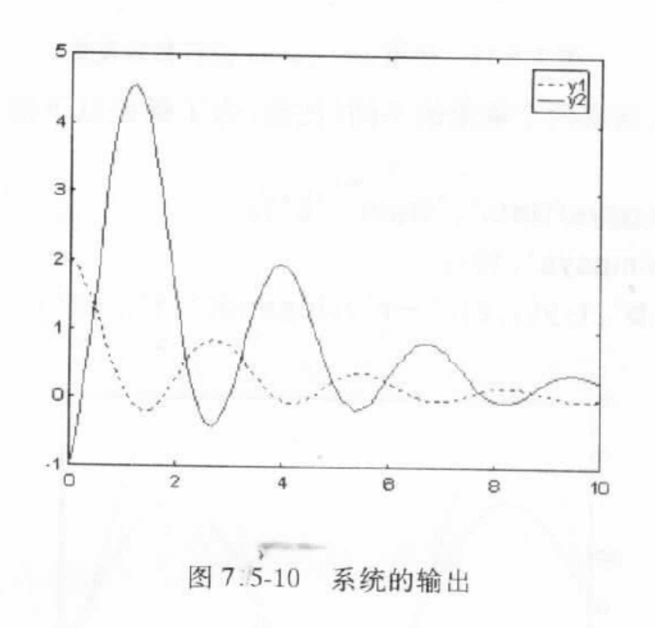


图 7.5-10 系统的输出

使用全局变量实现数据交换

在参数优化、灵敏度等计算中，常需要实现几个文件之间的数据交换，那么采用前面所说的预定义方式是不可行的。这时，可以采用全局变量来实现数据传送。定义全局变量的命令格式如下：

global a b c

在此，参数 a、b、c 被定义为全局变量。使用全局变量要注意：全局变量应在使用它们的所有命令文件、函数文件、工作内存中都加以定义，才能被共享。即，当其中某一个文件使全局变量数值发生改变后，新值马上传送到其他文件，当然也包括参与运行的框图模型。

使用 set _ param 指令传送数据

指令 set _ param 是专门设计来更改 SIMULINK 模块参数的。事实上，模块对话框中的参数设置都是靠这个指令实现的。

set _ param 函数的调用格式是

set _ param(Name, Parameter1, Value1, Parameter2, Value2, …)

其中，Name 是模块或系统名；Parameter 是待修改的参数名；Value 是新指定值。

【例 2】set _ param 运用示例。

(1) 构造图 7.5-11 所示的简单系统，文件名为 simpsys.m。

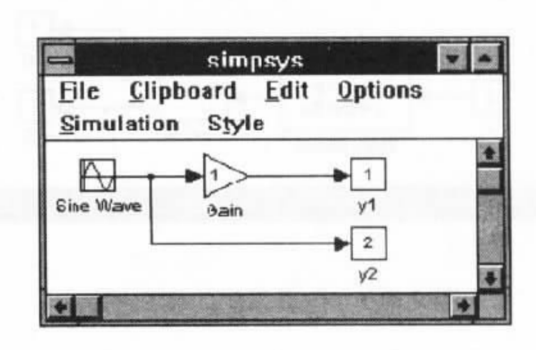


图 7.5-11　使用 set _ param 进行参数设置

(2) 运行以下指令，观察两个输出的不同(注意：为了保证以下指令运行，系统 simpsys.m 的框图窗必须被打开)。

```
set _ param('simpsys/Gain','Gain','2');
[t,x,y]=rk23('simpsys',10);
plot(t,y(:,1),':b',t,y(:,2),'-r');legend('y1','y2')
```

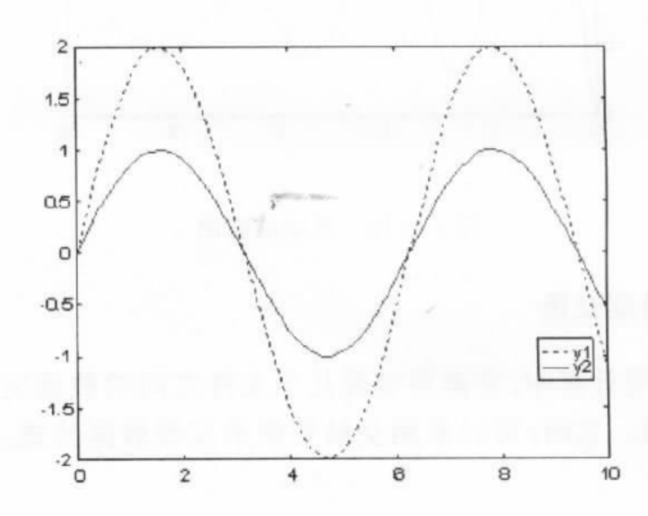


图 7.5-12　不同增益的信号输出

7.5.8　复杂模型的创建策略

SIMULINK 适用于构造非常复杂的动态系统模型(见演示实例)。创建复杂模型本身就是个富有挑战性的工作。为了使创建过程比较顺利，在此提出一些建议：

(1) 复杂系统的分解

假若，设计者感到系统太复杂，就应该把此系统分解成若干个子系统。

(2) 子系统的分别构作和测试

先单独构作各子系统，再单独测试之；然后，把几个子系统组合到一起，并进行组合后的测

试。经过几次测试、修改、再测试，直到整个系统全部创建成功。SIMULINK 为这种设计过程提供了各种方便的指令。

(3) 采用“重用”技术

在此，“重用”技术包括两方面的含义。一是，要充分通过对模块库中已有模块的改造来建立系统；二是，在设计和创建复杂系统过程中，要对共性建立通用模块。这不仅便于建模，也便于文件的归档管理和再使用。

第八章　MATLAB 的程序设计

用 MATLAB 语言写的简单程序在前面章节中已有所接触，本章将系统介绍如何用 MATLAB 语言进行程序设计。MathWork 公司将 MATLAB 语言称之为第四代编程语言，足见其简洁有效。MATLAB 的编程效率比常用的 BASIC、C、FORTRAN 和 PASCAL 等语言要高得多，而且维护容易。也正因为 MATLAB 程序的可读性很强，调试十分容易简单，所以本章不介绍 MATLAB 备有的 Debuge 指令。

8.1　M 文件的功能和特点

MATLAB 有两种常用方式：一种是直接交互的指令行操作方式；另一种是 M 文件的编程工作方式。在前一种工作方式下，MATLAB 被当作一种高级“数学演算和图视器”来使用。这种指令行交互操作方式如何在 MATLAB 视窗中工作、如何在 Notebook 中工作、如何与 SIMULINK 协调工作等内容，已经在前面几章作了比较充分的论述。本章着重介绍与后一种工作方式有关的内容。

MATLAB 是一个强有力的操作环境。它集中表现为 MATLAB 所提供的完整而易于使用的编程语言。从形式上讲，MATLAB 程序文件是一个 ASCII 码文件(标准的文本文件)，扩展名一律为 m(M 文件的名称由此而来)，用任何字处理软件都可以对它进行编写和修改。从特征上讲，MATLAB 是解释性编程语言，其优点是语法简单，程序容易调试，人机交互性强；缺点是由于逐句解释运行程序，故速度比编译型的慢。但是较慢运行速度仅明显表现在 M 文件初次运行时，因为 M 文件一经运行便变成代码存放在内存中。再次运行该文件时，MATLAB将直接从内存中取出代码运行，大大加快了运行速度。从功能上讲，M 文件大大地扩展了 MATLAB 的能力。Mathwork 公司推出的一系列工具箱 (Toolbox)就是明证。通过工具箱、MATLAB 才被应用到控制、鲁棒控制、信号处理、小波分析、系统辨识、图像处理、优化、样条分析、神经网络、金融财政等各个方面。而这些工具箱全部是由 M 文件构成的。从这点上来讲，如果不了解 M 文件，那么可以说，MATLAB 的能力仅应用了很小一部分。

由于 M 文件是解释性的程序语言，且以复数矩阵为基本运算单位，所以 M 文件无论从形式、结构、语法规则等方面都比一般的计算机语言简单、易写、易读得多。另外，MATLAB 本身是用 C 语言写的，M 文件的语法又与 C 语言十分相像，因此熟悉 C 语言的读者会轻松地掌握 MATLAB 的编程技巧。

8.2　M 文件的形式

M 文件有两种形式：命令文件(Script File)和函数文件(Function File)。这两种文件的扩

展名相同，都是“ m”。

当用户要运行的指令较多时，直接从键盘上逐行输入指令不是不可以，但显得比较麻烦；而命令文件则可以较好地解决这一问题。用户可以将一组相关命令编辑在同一个 ASCII 码命令文件中，运行时只需输入文件名字，MATLAB 就会自动按顺序执行文件中的命令。

函数文件是另一种形式的 M 文件，它的第一句可执行语句是以 function 引导的定义语句。在函数文件中的变量都是局部变量。

8.2.1 命令文件

命令文件中的语句可以访问 MATLAB 工作间(Workspace)中的所有数据。运行过程中，产生的所有变量均是全局变量。这些变量一旦生成，就一直保存在内存空间中，除非用户运用 Clear 指令将它们清除。

运行一个命令文件等价于从指令窗(Command Window)中按顺序连续运行文件里的指令。由于命令文件只是一串指令的集合，因此程序不需要预先定义，而只需按在指令窗中的指令输入顺序依次将指令编辑在命令文件中就可以。注意：不要忘记“ m”文件扩展名。

【例 1】计算所有小于 1000 的 Fibonnaci 数。

(1)用记事本(Notepad)编写以下内容。

fibno.m

```
% 计算小于 1000 的 Fibonnaci 数
f=[1,1];
i=1;
while f(i)+f(i+1)<1000
      f(i+2)=f(i)+f(i+1);
      i=i+1;
end
f,i
```

(2) 选择【文件】下拉菜单中的【保存】子项，将所写文件存放于磁盘中，并起名为 fibno m。

(3) MATLAB 指令窗中键入那文件名 fibno，运行结束，可在屏幕上看到以下内容。

```
f =
   Columns 1 through 12
    1    1    2    3    5    8   13   21   34   55   89  144
   Columns 13 through 16
   233   377   610   987
i =
   15
```

说明：

(1) 符号“%”引导的行是注释行，不予执行。

(2) 不需要用“end”作为 M 文件的结束标志。

(3) 若用户把文件 fibno m 存放在自己的工作目录(假如名为 d: \ mywork)上,那么在运行 fibno m 之前,应该先使 d: \ mywork 处于 MATLAB 的搜索路径上。最简单的方法是:在 MATLAB 指令窗中先运行 cd d: \ mywork。

(4) fibno m 运行后存放在内存中的变量,可以用 who 指令看到。

8.2.2 函数文件

如果 M 文件的第一行包含 function,此文件就是函数文件。每一个函数文件都定义一个函数。事实上,MATLAB 提供的函数指令大部分都是由函数文件定义的。这足以说明函数文件的重要。从使用的角度看,函数是一个"黑箱",把一些数据送进去,经加工处理,把结果送出来。从形式上看,函数文件区别于命令文件之处是:命令文件的变量在文件执行完后保留在内存中;而函数文件内定义的变量仅在函数文件内部起作用,当函数文件执行完后,这些内部变量将被清除。

【例 1】把上节例 1 中的命令文件改成计算小于任意自然数的函数文件,并运行之。

(1) 在字处理器中编写以下内容。

```
function f=ffibno(n)
%ffibno          计算 Fibonnaci 数的函数文件
%                f=ffibno(n)
%                n 可取任意自然数
%
%                1997.1 5 编
%                第 8 2 2 节示例 2
f=[1,1];
i=1;
while f(i)+f(i+1)<n
      f(i+2)=f(i)+f(i+1);
      i=i+1;
end
```

(2) 将文件 ffibno m 存盘。该文件定义了名为 fibno 的新函数。此函数的用法与别的 MATLAB 函数一样。

(3) 在 MATLAB 指令窗运行以下指令,便可求得小于 1000 的 Fibonnaci 数。

```
ffibno(1000)
ans=
    Columns 1 through 12
      1    1    2    3    5    8    13    21    34    55    89    144
    Columns 13 through 16
     233    377    610    987
```

说明:

(1) 第一行执行指令的作用:指明该文件是函数文件;定义函数名、输人参数和输出参数。函数名可以是 MATLAB 中任何合法的字符。输入和输出的参数根据实际需要设置,参数类型可以是数值,也可以是字符串。在本例中,输入参数是 n,输出参数是 f。在后面将进一步讨论如何进行函数参数的传递。

(2) 变量 ı 对函数文件 ffıbno m 来讲是局部的。当该函数被调用结束后,变量 ı 不再存在(如果变量 ı 在程序运行前就已经存在的话,程序运行后,它不会受到影响)。

(3) 在 M 文件前面,连续几行带符号“%”的注释行有两个作用:一是随 M 文件全部显示或打印时,直接起解释提示作用。二是供 help 指令在线查询用。

【例 2】在线查阅 ffıbno 函数的使用说明。

(1)在 MATLAB 指令窗中运行以下 help 指令,可得到帮助信息。

```
help ffibno
ffibno 计算 Fibonnaci 数的函数文件
      f = ffibno(n)
      n 可取任意自然数

      1997.1.5 编
```

(2) 利用 lookfor 指令对关键词的搜索,获取帮助信息。

```
lookfor Fibon
ffibno        计算 Fibonnaci 数的函数文件
```

说明:

(1) 该例 help 指令运行后所显示的是 M 文件注释语句中的第一个连续块。至于与第一连续块被空行所隔离的其他注释语句,将被 MATLAB 在线帮助系统忽略。

(2) 该例 lookfor 指令运行后,显示出 ffıbno 函数文件的第一行注释。一般地说,为了利用 MATLAB 对关键词的搜索功能,用户在编制 M 文件时,应在第一行注释中尽可能多地包含该函数的特征信息。

(3) 为了使 lookfor 指令能对用户目录上的 M 文件的关键词进行搜索,必须使用户目录处在 MATLAB 的搜索路径上。最简单的处理是·运行指令 path(path,'d:\mywork')。这里,d:\mywork 是假定的用户目录名。

8.3 程序结构

从理论上讲,只要有顺序、循环和分支三种基本程序结构,就可构成任何一种程序并完成相应的工作。与大多数计算机语言一样,MATLAB 有设计程序所必须的程序结构:(1)顺序结构;(2)循环结构;(3)分支结构。

在 MATLAB 语言中,循环有 whıle、for;分支结构由 ıf 语句表现。

MATLAB 虽然不像 C 语言那样具有丰富的控制结构,但是 MATLAB 自身的强大功能弥补了这个不足,使用户在编程中几乎感觉不到困难。MATLAB 语言是一种完善易用的高水平

矩阵编程语言。

8.3.1 顺序结构

MATLAB 的顺序结构实际上就是复合表达式构成的语句。复合表达式由分号或逗号隔离的几个表达式构成。当表达式后面接分号时，表达式的计算结果虽不显示但中间结果仍保留在内存中。若程序是命令文件，则程序运行完后，中间变量都予以保留；若程序是函数文件，那么程序运行完后，中间变量将被全部删除。

8.3.2 循环结构

在很多实际问题中会遇到许多有规律的重复运算，因此在程序中就需要将某些语句重复执行。一组被重复执行的语句称为循环体，每循环一次，都必须作出是继续重复或是停止的决定，这个决定所依据的条件称为循环的终止条件。MATLAB 语言提供了两种循环方式：for-end 循环和 while-end 循环。

for-end 循环

在许多情况下，循环条件是有规律地变化的，通常是把循环条件的初值，判别和变化放在循环的开头，这种形式就是 for 循环结构。概括讲，for-end 环的一般形式是：

```
for v = 表达式
     语句体
end
```

通常，表达式是一个向量，形如 m:s·n。s 是步长，它可以取整数、小数、正数或负数。该向量的元素被逐一赋给变量 v，然后执行语句体。在这个意义上，MATLAB 的 for 环与其他计算机语言没有什么区别。

【例 1】简单的 for 循环示例。

```
n=10;
for i=1:n
    x(i)=i;
end
x
x =
     1    2    3    4    5    6    7    8    9   10
```

说明：

(1) 在 1:n 这个 MATLAB 的冒号结构中，循环初值是 1，终值是 n，缺省步长为 1。循环体实现对 x 的前 n 个元素赋值。如果 n 小于 1，此环仍然合法，但环内的语句将不执行。如果 x 事先不存在或容量小于 n，那么超出的部分将自动补上。环可以嵌套。需要注意的是，每一个 for 必须与 end 相匹配，否则程序将出错。

(2) 本例只是为演示循环而设计。获得本例结果的更简单的指令是:x=1:10。

【例 2】循环的嵌套。

```
m=3;
n=4;
for i=1:m
    for j=1:n
     a(i,j)=1/(i+j-1);
    end
end
format rat
a
a =
       1        1/2       1/3       1/4
      1/2       1/3       1/4       1/5
      1/3       1/4       1/5       1/6
```

【例 3】运用非 1 步长的循环,产生 0~20 以内的偶数。

```
for i=0:2:20
    a(i/2+1)=i;
end
a
a =
      0    2    4    6    8    10    12    14    16    18    20
```

while-end 循环

While 环使语句体在逻辑条件控制下重复不确定次,直到循环条件为假。while 循环的一般形式是:

while 表达式
　　语句体
end

只要表达式的结果非零,语句体就重复执行。通常表达式几乎总是(1×1)的有理表达式,所以非零对应着“TRUE”。当表达式结果不是标量时,可以用 any,all 等函数处理。例如取表达式为 1 时,该循环将无休止进行。

【例 4】利用 while 环,解答:使 n! 是 100 位数的第一个 n 是什么?

```
n=1
while prod(1:n)<1e100
    n=n+1;
```

```
end
n =
    70
```

【例 5】矩阵指数的幂级数展开为 $e^A = I + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \cdots$。利用 while 环，求矩阵指数。

```
A=rand(3);
E=zeros(size(A));
F=eye(size(A));
K=1;
while norm(E+F-E,1)>0
     E=E+F;
     F=A*F/K;
     K=K+1;
end
E
expm(A)
E =
    1.5913     1.5455     1.1371
    0.5508     3.0230     1.5902
    0.9058     1.0966     1.5804
ans =
    1.5913     1.5455     1.1371
    0.5508     3.0230     1.5902
    0.9058     1.0966     1.5804
```

说明：

(1) 程序思想是：逐项求和，直到新的项加上去（在精度有限的计算机上）其和不再变化为止。在循环中，A 是给定的矩阵，E 是矩阵指数函数值，F 是展开式的项，K 是项数。循环体被重复直到 F 小到它被加到 E 上后并不使 E 变化。

(2) 调用 MATLAB 函数指令 expm(A) 所得结果与循环所得结果一致。

8.3.3　分支结构

在复杂的计算中常常需要根据表达式的情况（是否满足某些条件）确定下一步该作什么。MATLAB 的 if-else-end 提供了描述条件分支的结构。MATLAB 的 if 结构与其他计算机程序设计语言相似。if 语句可以分为三步来理解：

(1) 关键字 if 后的表达式反映了条件。表达式必须首先计算。

(2) 解释表达式的计算结果。MATAB 没有设计布尔量，而以变量值是否为零来表示假与真。计算结果为 0，就意味着“假”；反之，非零意味着“真”。

(3) 如果表达式值为真，则执行紧跟在其后的语句；否则转去执行另一分支。

分支结构的一般形式是：

```
if 表达式
      语句体 1;
else
      语句体 2;
end
```

说明：

(1) else 部分表示：当计算的表达式结果为假时，就做 else 后面语句体 2 的工作。当然，语句体 2 可以是复合语句或别的控制语句。

(2) 需要注意：if 语句嵌套时，if 和 else 必须对应，否则容易出错。

(3) 在 else 子句中也可嵌套 if 语句，这就形成了 elseif 结构。这种 else-if 结构实际上实现了一种多路选择。它在一定程度上可以代替一般高级编程语言中的 switch-case 语句。

【例 1】利用 MATLAB 的分支结构实现多路选择。

(1) 编写函数文件 bspline.m 计算 B-样条。

```
function f=bspline(x)
if x<0
      f=0;
elseif x<1
      f=x;
elseif x<2
      f=2-x;
else
      f=0;
end
```

(2) 利用函数文件，进行实数的分区间选择

```
bspline(-1)
ans =
       0
bspline(1.6)
ans =
    0.4000
bspline(2)
ans =
       0
```

【例 2】解数论问题。取任意整数，若是偶数，则用 2 除；否则乘 3 加 1。重复此过程，直到整数变为 1。有趣的是存在这样一个整数使问题无解，即循环将无止境的执行下去，能找到这个整数吗？

(1) 建立函数文件 collatz m。

```
function c=collatz(n)
% collatz
% Classic "3n+1" Problem from number theory
c = n;
while n > 1
      if rem(n,2)= =0
        n = n/2;
      else
        n = 3*n+1;
      end
      c = [c n];
end
```

(2) 利用函数文件,试探 n=9 所产生的情况。

```
collatz(9)
ans =
   Columns 1 through 12
      9   28   14   7   22   11   34   17   52   26   13   40
   Columns 13 through 20
     20   10    5   16    8    4    2    1
```

8.4 数据结构及全局变量

8.4.1 数据结构

与一般计算机语言相比,MATLAB 文件的数据结构比较简单。只有字符变量和数值变量两种。数值变量有整型和双精度型。它们的显示可以用指令窗菜单中【Options】中的【Numeric Format】命令来控制,也可以用指令 format 来控制。在程序设计中,不需要预定义数据类型,MATLAB 的程序解释器将自动根据所输入数据决定数据类型。

【例 1】字符变量的输入和检查。

```
a='This is my book'
isstr(a)
a =
This is my book
ans =
     1
```

说明：指令 isstr 用来检查变量是否是字符变量。返回 1，表示被检查变量是字符；返回 0，则表示被检查变量是数值。

【例 2】数值型变量的输入和检查。

```
a=sqrt(1000)
isstr(a)
a =
    31.6228
ans =
      0
```

【例 3】预定义变量可加快运算速度。

```
x=rand(10);
y=zeros(1,2000);
tic; for i=1:2000, z(i)=det(x); end; toc
tic; for i=1:2000, y(i)=det(x); end, toc
elapsed_time =
     4.3400
elapsed_time =
     2.1400
```

说明：

(1) 由于 MATLAB 文件是面向矩阵的语言，MATLAB 将任何一个变量都看成一个矩阵。如果变量是一个数，就认为是(1×1)的矩阵。通常情况下，变量不需要定义，但是预定义一个变量，可使程序在执行循环结构时的速度加快。

(2) 由于变量 z 没有预定义，所以程序解释器在每次循环中都重新定义 z 的维数。而 y 的预定义，使程序做循环时，省略定义维数这一步。为了仔细比较两者运行时间的区别，这里使用了指令 tic 和 toc 分别计算两个循环的运行时间，显然后者小于前者。

8.4.2 全局变量

全局变量用指令 global 定义。如运行指令 global X Y Z，就将 X、Y、Z 定义为全局变量。

正如前面所说的，函数文件的内部变量是局部的，与其他函数文件及 MATLAB 内存相互隔离。但是，如果在若干函数中，都把某一变量定义为全局变量，那么这些函数将公用这一个变量。全局变量的作用域是整个 MATLAB 工作空间，即全程有效。所有的函数都可以对它们进行存取和修改。因此，定义全局变量是函数间传递数据的一个手段。

值得指出·程序设计中，全局变量固然可带来某些方便，但却破坏了函数对变量的封装，降低了程序的可读性和可靠性。因而，在结构化程序设计中，全局变量是不受欢迎的。尤其当设计程序较大，子函数较多时，全局变量将给程序调试和维护带来不便，故而不提倡使用全局变量。如果一定要用全局变量，那么最好给它起一个特别的名字，以避免和其他变量混淆。

【例 1】利用全局变量，建一个计算 Fibonnaci 数的无参数传递的函数文件，并用它计算。

(1) 编写函数文件 gfibno m。

```
function f=gfibno
% gfibno m
% 利用全局变量 ndcba 计算 Fibonnaci 数
global ndcba
f=[1,1];
i=1;
while f(i)+f(i+1)<ndcba
      f(i+2)=f(i)+f(i+1);
      i=i+1;
end
```

(2) 在 MATLAB 指令窗中，运行以下指令：

```
global ndcba
ndcba=1000;
gfibno
ans =
   Columns 1 through 12
      1    1    2    3    5    8    13    21    34    55    89    144
   Columns 13 through 16
     233    377    610    987
```

8.5 程序流的控制

有关程序流控制的语句与函数，在前面章节中已零星见过。本节将系统介绍 echo, input, pause, break, keyboard 指令。

8.5.1 echo 指令

通常，M 文件执行时，文件中的指令不会显示在指令窗中。用 echo 命令可使文件指令在执行时可见。这对程序的调试和表演很有用。对命令文件和函数文件，echo 的作用稍微有些不同。

对命令文件，echo 的使用较为简单。其格式为·

echo on　　切换到显示其后所有被执行命令文件指令的状态。

echo off　　切换到其后所有被执行命令文件指令不被显示的状态。

echo　　实现被执行命令文件指令是否被显示的状态切换。

echo 的以上调用格式对函数文件不起作用。下面的 echo 调用格式对函数文件、命令文件都适用。具体如下：

echo FileName on　　使 FileName 指定文件的指令在执行中被显示出来。
echo FileName off　　终止显示 FileName 文件的执行过程。
echo FileName　　FileName 文件的执行过程是否被显示的切换开关。
echo on all　　显示其后所有被执行文件的过程。
echo off all　　使其后所有被执行文件的过程不被显示。

注意：当把 echo 运用于某一函数文件时，该文件将不被编译执行，而是被解释执行。这样，函数文件在执行过程中，每一行都可被观察到。由于这种解释执行不太有效，因而仅用于程序调试。

8.5.2 input、yesinput 指令

指令 input 提示用户从键盘输入数值、字符串或表达式，并接受该输入。下面是几种常用的格式：

(1) a=input('Please input a number:')

该指令运行后，将给出如下文字提示，并等待键盘输入：

Please input a number:

用户可以输入数字或表达式，也可以输入字符串(两端必须有单引号)，按【Enter】键确认后，该输入被赋给变量 a。

(2) a=input('Please input a string:','s')

该指令运行后，也给出提示：

Please input a string:

等待用户输入。在此提示后输入的任何内容(不管是数字还是字符)一律被当作字符串赋给变量 a。

(3) a=yesinput('Prompt:',Defaut,Possible)

yesinput 是一个智能输入指令。它带有缺省输入和输入的范围检查。该指令仅 MATLAB 4.2 版中有。它的用法较多，在此举例介绍两种。

【例 1】带输入数值范围检查的 yesinput 用法。

(1) 在 MATLAB 指令窗中键入

```
a=yesinput('Order of the filter',10,[0,12]);
```

(2) 该指令运行后，屏幕上有如下提示：

Order of the filter(10):

(3) 等待用户输入数据。如果只是按回车键，则默认输入值为 10；如果输入值大于 12 或小于 0 时，则认为输入无效，等待用户重新输入。

【例 2】带输入选项检查的 yesinput 用法。

(1) 在 MATLAB 指令窗中键入

```
color=yesinput('Color used on the plot','red','red|blue|green');
```

(2) 该指令运行后，屏幕上有如下提示

Color used on the plot(red):

(3) 等待用户输入数据。如果只是按回车键,则默认输入字符串为 red;如果输入字符串与选项内容不符,则认为输入无效,等待用户重新输入。

8.5.3 pause 指令

pause 指令使程序运行暂停,等待用户按任意键继续。pause 命令在程序调试以及需要看中间结果时特别有用。pause 的用法有两种:pause 暂停执行程序;pause(n)在继续执行前,暂停 n 秒。

8.5.4 keyboard 指令

Keyboard 与 input 一样有用。当程序遇到 keyboard 指令时,MATLAB 将暂停程序的运行并调用你的机器的键盘的命令进行处理。一旦处理完自己的工作后,键入 return, 然后按回车键,程序将继续进行。M 文件中含有它后,便于程序调试或在程序执行中修改变量。

8.5.5 break 指令

break 语句导致包含 break 指令的最内层 while、for、if 语句的终止。通过使用 break 语句,可不必等待循环的自然结束,而根据循环内部另设的某种条件是否满足,去决定是否退出循环,是否结束 if 语句。在很多情况下这样做是十分必要的。

【例 1】使用 break 解数学问题:求两个自然数,这两个数的和等于 100,且第一个数被 2 整除的商与第二个数被 4 整除的商的和为 36(这实际上是兔鸡同笼问题)。

```
i=1;
while 1
     if rem(100-i*2,4)==0 & (i+(100-i*2)/4)==36
        break;
     end
     i=i+1;
end
a1=i*2
a2=100-i*2
a1 =
     44
a2 =
     56
```

说明:

(1) 符号"&"表示布尔代数中的"与"运算。

(2) 在上述第三行中条件满足后，立即执行 break，使终止循环运作，并跳出循环体。如果没有这 break，本例循环将无休止进行。

8.6 字符与字符串

MATLAB 有强大的字符处理能力，特别是在引入符号计算(Symbolic Math)工具包以后。本节将讨论的是，字符处理在程序设计中的应用。对于编程语言来讲，字符处理是必不可少的。

在 MATLAB 中关于字符串有以下几点规则：

(1) 在 MATLAB 中所有字符串都用单引号界定后输入或赋值(yesinput 指令除外)。

如指令 s = 'Hello' 的运行结果是

```
s =
Hello
```

(2) 字符串的每个字符(空格也是字符)都是响应矩阵的一个元素。

如上述 s 变量是(1×5)的矩阵。这可用指令 size(s)查得。

(3) 字符以 ASCII 码贮存。用 abs 指令可看到字符的 ASCII 码值。

如运行 abs(s)，可得如下结果

```
ans =
   72    101    108    108    111
```

(4) 指令 setstr 实现 ASCII 码值向字符的转换。

(5) 字符变量也可以用方括号合并成更大的"串"。

例如，运行指令 s = [s, ' world ']，可得下面结果

```
s =
Hello world
```

(6) 用 eval 函数将字符变量转换为宏功能。eval(t)就是运行包含在 t 中的内容。

【例 1】用 eval 函数产生 5 阶的 Hilbert 矩阵。

```
n=5;
t='1/(i+j-1)';
a=zeros(n);
for i=1:n
    for j=1:n
        a(i,j)=eval(t);
    end
end
a
a =
    1.0000    0.5000    0.3333    0.2500    0.2000
```

```
0.5000    0.3333    0.2500    0.2000    0.1667
0.3333    0.2500    0.2000    0.1667    0.1429
0.2500    0.2000    0.1667    0.1429    0.1250
0.2000    0.1667    0.1429    0.1250    0.1111
```

【例 2】用 eval 传送函数。

```
i=25;
funname=['peaks(',int2str(i),')']
eval(funname);
funname =
peaks(25)
```

说明：

(1) eval 的宏文字功能可以把函数名传入函数文件。

(2) 构造字符串时应该注意方括号“[ ]”的使用。本例第二条语句产生的变量 funname 的值为 peaks(25)。

字符串操作函数

isstr 判断是否为字符。
blanks 空白字符。
deblank 移去空白字符。
eval 运行字符串。
strcmp 比较字符串。
findstr 从一个字符串中寻找是否包含另一个字符串。
strrep 用一个字符串代替另一个字符串。
upper 将字符串变为大写形式。
lower 将字符串变为小写形式。
abs 将字符串变为 ASCII 码值。
setstr 将 ASCII 码值变为字符串。
num2str 将数字变为字符串。
int2str 将整数变为字符串。
str2num 将字符串变为数字。
str2mat 将字符变为文本矩阵。
sprintf 将带格式的数字转变为字符串。
sscanf 将字符串转变为带格式的数字。
hex2num 将十六进制的字符串转变为 IEEE 浮点数。
hex2dec 将十六进制的字符串转变为十进制数。
dec2hex 将十进制数转变为十六进制字符串。

MATLAB 的字符操作函数与大部分高级编程语言特别是 C 语言的字符函数基本相同，用法也基本一样，这里不再做详细的介绍。

8.7 函数调用及变量传递

MATLAB 中的函数调用及变量传递比较复杂。而这两项工作又是编写高质量 M 文件所必不可少的。一个较大的计算任务可以分成若干个比较小的任务。这意味着，一个程序可由若干函数组成，并通过函数调用来实现控制转移和相互之间的数据传递。

8.7.1 函数调用

在 MATLAB 中，调用函数的常用形式是：

[输出参数 1，输出参数 2，]= 函数名(输入参数 1，输入参数 2，)

注意：函数调用时各参数出现的顺序，应该与函数定义时的顺序一样，否则出错。

函数调用可以嵌套，一个函数可以调用别的函数，甚至调用它自己(即递归调用)。

【例 1】给定两个实数 a、b 和一个正整数 n，给出 k=1，…，n 时的所有$(a+b)^n$和$(a-b)^n$(本例限定不大于 10)。

(1) 建立函数文件 power m。

```
function [out1,out2]=power(a,b,n)
% Power m 计算(a+b)^n 和 (a-b)^n
out1=(a+b)^n;
out2=(a-b)^n;
```

(2) 建立调用上述函数文件的命令文件 examp1 m。

examp1.m

```
a=input('Please input a=:');
b=input('Please input b=');
addpow=zeros(1,10);
subpow=zeros(1,10);
for k=1:10
    [addpow(k),subpow(k)] = power(a,b,k);
end
addpow
subpow
```

说明：在本例中，命令文件 exampl1 对函数 power 每做一次调用，就传入三个参数 a、b、k，送出两个结果 addpow 和 subpow。

【例 2】用递归调用形式计算 n 的阶乘。

(1) 编写递归调用函数文件 factor m

```
function f=factor(n)
```

```
% factor.m          计算 n 的阶乘
if n==1
      f=1;
      return;
else
      f=n*factor(n-1);
      return;
end
```

(2) 运行程序函数文件

```
factor(6)
ans =
    720
factor(10)
ans =
       3628800
```

说明：递归函数是一类算法编程的最直接方式，在程序设计中有重要地位。递归调用可使程序设计方便、简洁、易读。这从又一个侧面说明，MATLAB 处理函数调用功能的强大。

8.7.2 参数传递

MATLAB 在函数调用上的有一个与众不同之处：函数所传递参数数目的可调性。凭借这种特性，一个函数可完成多种功能。

传递参数数目的可调性来源于如下两个 MATLAB 永久变量(Permanent Variable)：

nargin　　　　函数体内的 nargin 给出调用该函数时的输入参数数目。

nargout　　　　函数体内的 nargout 给出调用该函数时的输出参数数目。

只要在函数文件内包含这两个变量，就可准确地知道该函数文件被调用时的输入输出数，从而决定函数如何进行处理。

【例 1】以 MATLAB 提供的“彗星线”绘制函数 comet.m 为例说明 nargin 的用法（该函数不带输出参数，但有三个可选的输入参数）。

```
function comet(x, y, p)
if nargin == 0,                         % 没有输入参数则 nargin=0
   x = -pi:pi/200:pi;                   % 在程序中定义变量 x,y,p
   y = tan(sin(x))-sin(tan(x));
   p = 0.1;
else
   if nargin < 2, y = x; x = 1:length(y); end
                                        % 有一个输入参数则 nargin=1
   if nargin < 3, p = 0.10; end         % 有两个输入参数则 nargin=2
```

```
end
clf
axis([min(x(finite(x))) max(x(finite(x))) min(y(finite(y))) ...
max(y(finite(y)))])
% Cyan circle head, yellow line body, and magenta line tail
head=line('color','c','linestyle','o','erase','xor','xdata',x(1),'ydata',y(1));
body=line('color','y','linestyle','-','erase','none','xdata',[],'ydata',[]);
tail = line('color','m','linestyle','-',
'erase','none','xdata',[],'ydata',[]);
m = length(x);
k = round(p * m);
% Grow the body
for i = 2:k+1
   j = i-1:i;
   set(head,'xdata',x(i),'ydata',y(i))
   set(body,'xdata',x(j),'ydata',y(j))
   drawnow
end
% Primary loop
for i = k+2:m
   j = i-1:i;
   set(head,'xdata',x(i),'ydata',y(i))
   set(body,'xdata',x(j),'ydata',y(j))
   set(tail,'xdata',x(j-k),'ydata',y(j-k))
   drawnow
end
% Clean up the tail
for i = m+1:m+k
   j = i-1:i;
   set(tail,'xdata',x(j-k),'ydata',y(j-k))
   drawnow
end
```

说明:

(1) 该函数的三个输入参数中,向量 x 和 y 定义了“彗星”的运动轨迹,变量 p 定义了“彗星”的长度。

(2) 运行不带输入参数的命令 comet 时,函数内部就自己定义 x、y、p,形成它的自我演示。

(3) 运行带一个输入参数的命令 comet(y)时,函数在采用 y 向量的同时,再定义相应的 x 和 p。

(4) 运行带两个输入参数的命令 comet(x,y)时,函数采用缺省的 p 值。

(5) 虽然输入参数个数可变,但参数的输入顺序不能改变。

8.8 数据的输入与输出

MATLAB 不但是一个自包容的高效工作环境,而且能与其他外部应用程序进行数据交换。因有关 MATLAB 数据输入与输出的内容比较庞杂,本节只能扼要介绍。更详细的描述参见 MATLAB 用户手册。

8.8.1 数据的输入

输入数据的方法很多,究竟选择那种方法取决于:数据的多少,数据的可读性,数据形式等。下面罗列几种方法以供选择:

(1) 按元素列表直接送入数据

如果数据不多(比方 10~15 个),那么利用方括号[ ],直接从键盘输入最方便。数据较多时,这种方法不好,因为一旦有错,无法修改。

(2) 用 M 文件产生数据

利用文本编辑产生一个命令 M 文件。由该文件直接把按元素列表方式的数据引入 MATLAB 工作内存。当数据不是机器可读的或是已经被打印出来的情况下,这个方法最合适。其优点在于,可利用文本编辑器修改数据。

(3) 从 ASCII 码平面文件装载数据

平面文件(Flat File)相对关系数据库(Relational Database)而言。数据以 ASCII 码形式存贮,带回车结尾的固定长度行,数码以空格分隔。该平面文件可用命令 load 直接读入 MATLAB,其结果放在以文件名为名的变量中。

(4) 利用 fopen、fread 及 MATLAB 其他底层 I/O 指令读数据

该方法用于读取其他外部应用程序以自己格式建立的数据。

(5) 生成 MEX 文件去读取数据

在读取其他应用数据的外部子程序现成的情况下,采用这种方法。

(6) 通过 MAT 文件读取数据

先用 C 或 FORTRAN 编一个程序把应用数据转换成 MAT-文件形式,然后再把此 MAT-文件读入 MATLAB。

8.8.2 数据的输出

MATLAB 输出数据的方式有以下多种选择:

(1) 利用 diary 指令输出数据

在 MATLAB 指令窗里运行 diary 指令可产生一个日记文件。它将记录此后 MATLAB 指令窗中显示的内容(包括指令、中间运算结果、最终结果等)。该文件可以用文本编辑器进行编辑整理。该日记不仅提供数据,而且还为以后用户写总结文件或报告提供便于使用的原始素

材。

(2) 利用 Notebook 获取数据

指令 dıary 产生的日记文件的缺陷是：内容比较乱；不包含 MATLAB 运作中产生的数据图形。在 Notebook 环境下工作所产生的 M-book 文件，不仅文字质量高、版面规范漂亮，而且 MATLAB 的指令、数据、图形尽纳其中。更可贵的是 M-book 中的 MATLAB 指令可随时运行，随时修改，MATLAB 工作内存中的数据随之改变，M-book 将始终保持文件中的数据、图形与指令相一致。从这 M-book 文件(是 Word 中的一种特殊 doc 文件)可产生专业质量的印刷品。

(3) 利用指令 save 输出数据

save 可以将当前 MATLAB 内存中的几个或全部变量存到文件中去(详见 8 8 3 节)。

(4) 利用 fopen、fwrıte 及其他底层 I/O 指令输出特殊格式的数据

当其他外部应用中需要某种格式数据时，要采用此法。

(5) 用 MEX 文件写数据

如若满足应用需要格式数据的某子程序已经存在，那么采用这种方法。

8.8.3 save 和 load 指令使用

这两个指令在 MATLAB 与外部环境交换数据时最常用。Save 指令将 MATLAB 工作内存中的变量存入磁盘；load 指令则把磁盘上的数据送入内存。

save 的常用方法如下：

(1) save

将 MATLAB 工作内存中所有变量以二进制格式存入 matlab mat 缺省文件。当数据以二进制格式存储时，其存储的数据的精度依赖于数据的多少，数据越多精度越差。对维数不超过 10000 个元素的矩阵，每个元素用 8 个字节。

(2) save dfıle

将工作内存中所有变量以二进制格式存入 dfıle mat 文件，扩展名自动产生。

(3) save dfıle x y

只把变量 x 和 y 以二进制格式存入 dfıle mat 文件，扩展名自动产生。

(4) save dfıle dat x y -ascıı

将变量 x 和 y 以 8 位 ASCII 码形式存入 dfıle dat 文件。

(5) save dfıle dat x y -ascıı -double

将变量 x 和 y 以 16 位 ASCII 码形式存入 dfıle. dat 文件。

load 指令的应用格式比较简单，具体如下：

load　　　　把(缺省)磁盘文件 matlab. mat 内容读入内存。

load dfıle　　把磁盘文件 dfıle mat 内容读入内存。

load dfıle dat　把磁盘文件 dfıle dat 内容读入内存。

8.8.4 不同平台间的数据交换

MATLAB 的 M 文件(包括命令文件、函数文件)是标准的 ASCII 码文本文件,它们的格式与机器的类型无关。MATLAB 的每个 MAT 文件(二进制数据文件)在文件头上都有一个机器特征码。当 MAT 文件被装载时,这机器特征码首先被识别。当发现该码代表其他机种时,就会自动作必要的转换。因此,作为 MATLAB 的两大应用文件(M 文件和 MAT 文件)都可以直接在不同机器间相互交换。

到目前为止的各 MATLAB 版本,自身都不具备进行不同机器间通讯的能力。因此,无论是通讯交换二进制文件还是 ASCII 文件,都需要借助其他通讯软件,如 FTP、NFS、Kermit 等。而且,在使用这些通讯软件时要格外小心,任何设置错误都会导致数据传输的失败。据 MATLAB 开发销售商自己预告,新的 MATLAB 5.0 版将着手解决远程通讯问题。

附录 MATLAB主要函数指令表

A0 主要函数指令分类

general	通用指令
ops	运算符和特殊算符
elfun	基本数学函数
elmat	基本矩阵和矩阵操作
strfun	字符串函数
matfun	矩阵函数和数值线性代数
datafun	数据分析和傅里叶变换
polyfun	多项式与插补函数
funfun	非线性数值功能函数
plotxy	二维图形
plotxyz	三维图形
graphics	通用绘图函数
color	色彩控制和光照模式
specfun	特殊数学函数
specmat	特殊矩阵
lang	语言结构和调试指令
iofun	底层文件输入/输出函数
sparfun	稀疏矩阵函数
sounds	声音处理函数
dde	动态数据交换函数
local	主启动文件
demos	演示函数

A1 常用指令(General Purpose Commands)

A1.1 管理指令和函数(Managing Commands and Functions)

demo	演示程序	2 4 3
help	在线帮助指令	2 5 1，4 12 1，6 5 3，8 2 2
lookfor	关键词检索	2 5 2，4 12 1，6 5 3；8 2 2
path	控制 MATLAB 的搜索路径	2 6 2，8 2 2
pathtool	路径管理器	2 3 2，2 6 2，6 5 1
type	显示文件内容	2 4 2；4 12 1，6 5 3
what	列出当前目录上的 m-文件、mat-文件和 mex-文件	2 5 3
which	确定指定函数和文件的位置	2 5 3，6 5 3

A1.2 变量和内存空间的管理(Managing Variables and the Workspace)

clear	从内存中清除变量和函数	2 4 2；2 4 4，2 4 6；8 2 1
disp	显示矩阵和文字内容	2 4 2
length	确定向量的长度	3 13 1
load	从磁盘中调入数据变量	3 1 5；8 8 3
pack	合并工作内存中的碎块	2 4 2
save	把内存变量存入磁盘	3 1 5，8 8 3
size	确定矩阵的维数	3 7 2，3 8 3，3 8 4，3 10 2，3 12 3
who	列出工作内存中的变量名	2 4 6，6 5 3
whos	列出工作内存中的变量细节	2 4 6；6 5 3
workspace	工作内存浏览器	6 5 2，8 2 1

A1.3 调用操作系统和文件处理(Working with Files and the Operating System)

cd	改变当前工作目录	2 4 2；2 6 2；8 2 1
delete	删除文件	
diary	储存 MATLAB 指令窗操作内容	8 8 2
dir	列出的文件	2 4 2
edit	矩阵编辑器	3 1 2
getenv	给出环境值	
!	执行外部应用程序	

A1.4 指令窗控制(Controlling the Command Window)

clc	清除指令窗口	2 4 2
echo	显示命令文件指令的切换开关	2 4 2，8 5 1
format	设置数据输出格式	3 1 6，8 4 1
home	光标返回行首	
more	命令窗口分页输出的控制开关	

A1.5 MATLAB 的启动和终止(Starting and Quitting from MATLAB)

quit	退出 MATLAB	2 4 2
matlabrc	MATLAB 的主启动文件	2 2 2，2 3 2
startup	启动 MATLAB 时的自动执行 M 文件	

A1.6 通用信息查询(General Information)

hostid	MATLAB 服务中心识别号
info	关于 MATLAB 和 MathWorks 公司的信息
subscribe	MATLAB 用户注册
whatsnew	未入档的新特征信息
ver	MATLAB, SIMULINK 和 TOOLBOX 的版本信息

A2 运算符和特殊算符(Operators and Special Characters)

A2.1 常用符 (Operators and Special Characters)

+	加	(arith)	2 4 7；3 2 1
−	减	(arith)	2 4 7，3 2 1
*	矩阵乘	(arith)	2 4 7，3 2 1；3 2 2
*	数组乘	(arith)	3 2 1；3 2 2
^	矩阵乘方	(arith)	2 4 7；3 2 1，3 4 1，3 4 2
^	数组乘方	(arith)	3 2 1，3 4 1，3 4 2

\	反斜杠或左除	(slash)	3 2 1, 3 3 1, 3 3 2, 3 3 3, 3 7 2
/	斜杠或右除	(slash)	2 4 7, 3 2 1; 3 3 1
/	数组除	(slash)	3 2 1, 3 3 4
kron	张量积		3 8 5
	冒号	(colon)	3 8 1, 3 8 6
( )	圆括号	(paren)	3 8 2
[ ]	方括号	(paren)	3 1 1, 3 8 3, 4 2 2, 8 6
	小数点	(punct)	
	续行号	(punct)	2 4 5; 3 1 4
,	逗号	(punct)	3 1 1
,	分号	(punct)	2 4 5; 3 1 1, 4 2 2
%	百分号	(punct)	3 1 1; 3 1 4, 8 2 1; 8 2 2
!	惊叹号	(punct)	
'	转置号和引号	(punct)	3 2 1; 4 2 1, 8 6
=	赋值符号	(punct)	2 4 4
= =	等号	(relop)	3 6 1
&	逻辑与	(relop)	3 6 2
|	逻辑或	(relop)	3 6 2
~	逻辑非	(relop)	3 6 2
xor	逻辑异或		3 6 2; 3 6 3

【注】本表第三栏括号中的字符供在线求助时 help 指令引述用。

A2.2 逻辑操作(Logical Characteristics)

all	全非零元向量为真	3 6 3
any	有非零元的向量为真	3 6 3
exist	检查变量或函数是否被定义	3.6 3
find	找出非零元素 1 的下标	3 6 3
finite	若是有限数则为真	3 6 3
isempty	若是空矩阵则为真	3 6 3
isglobal	若是全局变量则为真	3 6 3
isinf	若是无穷大则为真	3 6 3
isnan	若为非数则为真	3 6 3
isreal	若是实数矩阵则为真	3 6 3
issparse	若是稀疏矩阵则为真	3,6 3, 3 12 3
isstr	若是字符串则为真	3 6 3, 8 4 1

A3 基本数学函数(Elementary Math Functions)

A3.1 三角和超越函数(Trigonometric and Hyperbolic)

acos	反余弦	3 5 1
acosh	反双曲余弦	3 5 1
acot	反余切	3 5 1
acsc	反余割	3 5 1
acsch	反双曲余割	3 5 1
asec	反正割	3 5 1

asech	反双曲正割	3 5 1
asin	反正弦	3 5 1
asinh	反双曲正弦	3 5 1
atan	反正切	3 5 1
atanh	反双曲正切	3 5 1
atan2	四象限反正切	3 5 1
cos	余弦	3 5 1
cosh	双曲余弦	3 5 1
cot	余切	3 5 1
coth	双曲余切	3 5 1
csc	余割	3 5 1
csch	双曲余割	3 5 1
sec	正割	3 5 1
sech	双曲正割	3 5 1
sin	正弦	3 5 1
sinh	双曲正弦	3 5 1
tan	正切	3 5 1
tanh	双曲正切	3 5 1

A3.2 指数和对数函数(Exponential and Logarithm)

exp	指数	3 2 1,3 5 1,3 5 3
log	自然对数	3 2 1;3 5 1,3 5 3
log10	常用对数	3 5 1
sqrt	平方根	3 2 1,3 5 1,3 5 3

A3.3 复数函数(Complex)

abs	绝对值	3 5 1 ; 8 6
angle	相角	3 5 1 , 3 11
conj	复数共轭	3 2 1 , 3 5 1
imag	复数虚部	3 5 1
real	复数实部	3 5 1 ; 3 9 1

A3.4 数值处理(Numeric)

ceil	朝正无穷大方向取整	3 5 1
fix	朝零方向取整	3 5 1
floor	朝负无穷大方向取整	3 5 1
rem	求余数	3 5 1
round	四舍五入取整	3 5 1
sign	符号函数	3 5 1

A4 基本矩阵函数和操作(Elementary Matrices and Matrix Manipulation)

A4.1 基本矩阵(Elementary Matrices)

eye	单位阵	3 8 4
linspace	线性等分向量	3 8 1 ; 3 8 4
logspace	对数等分向量	3 8 1 ; 3 8 4
meshgrid	用于三维曲面的分格线坐标	3 8 4 ; 5 3 1 ; 5 3 5

ones	全 1 矩阵	3 8 4
rand	均匀分布随机阵	3 2 1 ;3 3 1 ; 3 8 4
randn	正态分布随机阵	3 8 4
zeros	全零矩阵	3 8 4 ; 8 4 1

A4.2 特殊变量和常数(Special Variables and Constants)

ans	最新表达式的运算结果	2 4 5
computer	计算机类型	
eps	浮点相对误差	2 4 6 , 2 4.7
flops	浮点运算次数	
i, j	虚数单位	2 4 6 ; 2 4 8 , 3 1 1 , 4 4 3 , 4 4 4
inf	无穷大	2 4 6 , 3 3 1 , 3 6 3
isieee	计算机采用 IEEE 算法则为真	
NaN	非数	2 4 6 ; 3 6 3 , 5 3 1
nargin	函数输入宗量的个数	8 7 2
nargout	函数输出宗量的个数	8 7 2
pi	3 1415926535897	2 4 6 ,3 1 1
realmax	最大浮点数	
realmin	最小正浮点数	
version	MATLAB版本	
why	一般问题的简明答案	

A4.3 时间和日期(Time and Dates)

clock	时钟	
cputime	MATLAB 占用 CPU 时间	
date	日期	
etime	用 CLOCK 计算的时间	
tic	秒表起动	3 3.1 , 8 4 1
toc	秒表终止和显示	3 3 1 , 8 4 1

A4.4 矩阵操作(Matrix Manipulation)

diag	创建对角阵,抽取对角向量	3 1 3 , 3 8 5 ,3 8 6
fliplr	矩阵的左右翻转	3 8 6
flipud	矩阵的上下翻转	3 8 6
reshape	矩阵变维	3 1 3 , 3 8 6
rot90	矩阵逆时针旋转 90 度	3 8 6
tril	抽取下三角阵	3 8 6
triu	抽取上三角阵	3 8 6
	矩阵的援引和重排	3 8 1 , 3 8.6

A5 字符串函数(Character String Functions)

A5.1 通用字符串函数(General String Functions)

abs	把字符串变成 ASCII 码值	3 5 1 ; 8 6
blanks	空格符号	8 6
deblank	删除最后的空格	8 6
eval	执行串形式的 MATLAB 表达式	8 6

isstr	若是字符串则为真	363;841
setstr	把 ASCII 码值变成字符串	86
strings	MATLAB 中的字符串	
str2mat	把单个串变成文本矩阵	86

A5.2 串比较(String Comparison)

isletter	串中是字母则为真	
isspace	串中是空格则为真	
findstr	在一个串中寻找一个子串	86
lower	把字符串变成小写	86
strcmp	比较字符串	86
strrep	用另一个串代替一个串中的子串	86
strtok	删除串中的指定子串	
upper	把字符串变成大写	86

A5.3 字符串与数值间的转换(String to Number Conversion)

int2str	把整数转换为串	86
num2str	把浮点数转换为串	86
sprintf	按格式把数字转换为串	86
sscanf	按格式把串转换为数字	86
str2num	把串转换为浮点数	86

A5.4 十六进制数与十进制数值间的转换(Hexadecimal to Number Conversion)

dec2hex	把 10 进制整数变成 16 进制串	86
hex2dec	把 16 进制串变成 10 进制整数	86
hex2num	把 16 进制串变成 IEEE 浮点数	86

A6 矩阵函数和数值线性代数(Matrix Functions - Numerical Linear Algebra)

A6.1 矩阵分析(Matrix Analysis)

cond	矩阵条件数	352
det	行列式的值	352
norm	矩阵或向量范数	352
null	零空间	37
orth	值空间	37
rank	秩	352
rcond	LINPACK 逆条件数	352
rref	转换为行阶梯形	
trace	迹	352

A6.2 线性方程(Linear Equations)

chol	Cholesky 分解	37
inv	矩阵的逆	331
lscov	已知协方差的最小二乘解	
luLU	分解	37
nnls	非负最小二乘解	
pinv	伪逆	372
qr	QR 分解	37

\ 、/	解线性方程	3 2 1，3 3 1，3 3 2，3 3 3，3 7 2

A6.3 特性值与奇异值(Eigenvalues and Singular Values)

balance	改善特征值精度的平衡刻度	
cdf2rdf	复数对角型转换到实块对角型	3 7,3 7 1
eig	矩阵特征值和特征向量	3 5 2;3 7;3 7 1
hess	Hessenberg 矩阵	3 7
poly	特征多项式	3 9 1
polyeig	多项式特征值问题	
qz	广义特征值	3 7
rsf2csf	实块对角型转换到复数对角型	3 7 1
schur	Schur 分解	3 7
svd	奇异值分解	3 5 2;3 7 2

A6.4 矩阵函数(Matrix Functions)

expm	矩阵指数	3 2 1，3 5 2，3 5 3，8 3 2
expm1	矩阵指数的 Pade 逼近	3 5 2
expm2	用泰勒级数求矩阵指数	3 5 2
expm3	通过特征值和特征向量求矩阵指数	3 5 2
funm	计算一般矩阵函数	3 5 2，3 5 3
logm	矩阵对数	3 5 2,3 5 3
sqrtm	矩阵平方根	3 5 2,3 5 3

A7 数据分析和傅里叶变换(Date Analysis and Fourier Transform Functions)

A7.1 基本运算(Basic Operations)

cumprod	元素累计积	3 10 1
cumsum	元素累计和	3 10 1，3 13 1
max	最大值	3 10 1
mean	平均值	3 10 1
median	中值	3 10 1
min	最小值	3 10 1
prod	元素积	3 10 1
sort	由小到大排序	3 10 1
std	标准差	3 10 1
sum	元素和	3 8 4，3 12 3
trapz	梯形数值积分	3 13 1

A7.2 有限差分(Finite Differences)

del2	五点离散 Laplacian	3 10 4
diff	差分和近似微分	3 10 4
gradient	梯度	3 10 4

A7.3 向量运算(Vector Operations)

cross	向量叉积	
dot	向量内积	3 5 2

A7.4 相关(Correlation)

corrcoef	相关系数	3 10 2

cov	协方差矩阵	3 10 2
subspace	子空间的角度	

A7.5 滤波和卷积(Filtering and Convolution)

conv	卷积和多项式相乘	3 9 2,3 11
conv2	二维卷积	
deconv	解卷和多项式相除	3 9 2;3 11
filter	一维数字滤波器	3 11 2
filter2	二维数字滤波器	

A7.6 傅里叶变换(Fourier Transforms)

abs	幅值	3 11 , 8 6
angle	相角	3 5 1 ; 3 11
cplxpair	复数阵成共轭对形式排列	
fft	快速离散傅里叶变换	3 11 1
fftshift	重排 fft 和 fft2 的输出	
fft2	二维离散傅里叶变换	
ifft	离散傅里叶反变换	3 11 1
ifft2	二维离散傅里叶反变换	
nextpow2	最近邻的 2 的幂	
unwrap	相位角 360 度线调整	

A8 多项式与插补函数(Polynomial and Interpolation Functions)

A8.1 多项式(Polynomials)

conv	多项式相乘	3 9 2
deconv	多项式相除	3 9 2
poly	由根创建多项式	3 9 1
polyder	多项式微分	
polyfit	多项式拟合	3 9 3
polyval	求多项式的值	3 9 3
polyvalm	求矩阵多项式的值	3 9 3
residue	求部分分式表达	3 9 3
roots	求多项式的根	3 9 3

A8.2 数据插补(Data Interpolation)

griddata	三维分格点数据	
interpft	利用 FFT 方法一维插补	
interp1	一维插补	
interp2	二维插补	

A8.3 样条插补(Spline Interpolation)

spline	三次样条插补	
ppval	计算分段多项式	

A9 非线性数值功能函数(Function Functions - Nonlinear Numerical Methods)

fmin	单变量函数最小值	3 13 2

fmins	多变量函数最小值	3 13 2
fplot	画函数曲线图	5 2 2
fzero	单变量函数的零点	3 13 2
ode23	低阶法解微分方程	3 13 3
ode23p	解方程并画出解的曲线	3 13 3
ode45	高阶法解微分方程	3 13 3
quad	低阶法数值积分	3 13 1
quad8	高阶法数值积分	3 13 1

A10 二维图形函数(Two Dimensional Graphics)

A10.1 基本平面图形(Elementary X-Y Graphs)

fill	平面多边形填色	5 2 4
loglog	双对数刻度曲线	
plot	直角坐标下线性刻度曲线	5 1 1
semilogx	X 轴半对数刻度曲线	
semilogy	Y 轴半对数刻度曲线	

A10.2 特殊平面图形(Specialized X-Y Graphs)

bar	直方图	5 2 1
compass	从原点出发的复数向量图	5 2 1
comet	彗星状轨迹图	5 9 1
errorbar	误差棒棒图	5 2 1
feather	从 X 轴出发的复数向量图	5 2 1
fplot	函数曲线图	5 2 2
hist	统计频数直方图	3 10 3
polar	极坐标曲线图	5 2 1
rose	统计频数扇形图	3 10 3
stairs	阶梯形曲线图	5 2 1
stem	火柴杆图	5 2 1

A10.3 二维图形注释(Graph Annotation)

grid	画坐标网格线	5 5 3
gtext	用鼠标在图上标注文字	5 5 2
legend	图例说明	3 9 3；3 13 3；5 5 1
text	在图上标注文字	5 5 2
title	图形标题	5 5 1
xlabel	X 轴名标注	5 5 1
ylabel	Y 轴名标注	5 5 1

A11 三维图形函数(Three Dimensional Graphics)

A11.1 线、面填色指令(Line and Area Fill Commands)

comet3	三维彗星动态轨迹线图	5 9 1
fill3	三维曲面多边形填色	5 2 4
plot3	三维直角坐标曲线图	5 2 3

A11.2 三维数据的等高线和其他二维表现(Contour and Other 2-D Plots of 3-D Data)

clabel	给等高线加标注	5 3 4
contour	等高线图	5 3 4
contourc	等高线计算	5 3 4
contour3	三维等高线图	5 3 4
pcolor	用颜色反映数据的伪色图	5 3 3，5 7 3
quiver	矢量场图	3 10.4，5 3 5

A11.3 曲面与网线图(Surface and Mesh Plots)

mesh	三维网线图	5 1 2
meshc	带等高线的三维网线图	5 3 1
meshz	带零基准面的三维网线图	5 3 1
surf	三维表面图	5 3 2，5 3 3，5 7 3，5 11 2，5 11 3
surfc	带等高线的三维表面图	5 3 2
surfl	带光照的三维表面渲染图	5 3 2，5 6 6
waterfall	瀑布水线图	5 3 1，5 11 3

A11.4 内部视图(Volume Visualization)

slice	切片图	5 4

A11.5 图的表现(Graph Appearance)

axis	轴的刻度和表现	5 6 4
caxis	(伪)颜色轴刻度	5 11 3
colormap	设置色图	5 7 1，5 7 2，5 7 3，5 7 5，5 11 3
hidden	消隐	5 3 1
shading	图形渲染模式	5 3 1，5 7 4
view	设定 3-D 图形观测点	5 6 5，5 11 3
viewmtx	观测点转换矩阵	

A11.6 三维图的注释(Graph Annotation)

grid	画坐标网格线	5 5 3
gtext	用鼠标在图上标注文字	5 5 2
text	在图上标注文字	5 5 2
title	图形标题	5 5 1
xlabel	X 轴名标注	5 5 1
ylabel	Y 轴名标注	5 5 1
zlabel	Z 轴名标注	5 5 1

A11.7 其他三维图形对象(3-D Objects)

cylinder	圆柱面	5 3 6
sphere	球面	5 3 6

A12 通用图形函数(General Purpose Graphics Functions)

A12.1 图形窗的产生和控制(Figure Window Creation and Control)

clf	清除当前图	2 4 2，5 6 1，5 6 2，5 11 3
close	关闭图形	
figure	打开或创建图形窗口	5 6 1
gcf	获得当前图的柄	5 11 3

A12.2 轴的产生和控制(Axis Creation and Control)

axes	在任意位置创建轴	5 11 2，5 11 3
axis	轴的控制	5 6 4,5 7 5
caxis	控制色轴的刻度	5 11 3
cla	清除当前轴	5 7 3,5 11 3
gca	获得当前轴的柄	5 11 3，6 4 2
hold	图形的保持	5 6 3，5 11 3
subplot	创建子图	5 6 2

A12.3 图形对象的句柄(Handle Graphics Objects)

axes	创建轴	5 11 1，5 11 2
figure	创建图形窗口	5 11 1，5 11 2
image	创建图象	5 7 5，5 11 1，5 11 2
line	创建线	5 11 1，5 11 2
patch	创建块	5 11 1，5 11 2
surface	创建面	5 11 1，5 11 2
text	创建图形中文本	5 11 1；5 11 2
uicontrol	用户使用界面控制	5 11 1，5 11 2
uimenu	用户使用菜单控制	5 11 1，5 11 2

A12.4 图形句柄操作(Handle Graphics Operations)

delete	删除对象及文件	
drawnow	屏幕刷新	5 11 4
findobj	用规定的特性找寻对象	
gco	获得当前对象的柄	
get	获得对象特性	5 11 3
newplot	下一个新图	
reset	重设对象特性	5 11 3
set	建立对象特性	5 11 3，6 4 2

A12 5 打印和存储(Hardcopy and Storage)

capture	当前图的屏捕捉	
orient	设置走纸方向	5 11 3
print	打印图形或把图存入文件	5 10 1,5 10 2
printopt	打印机设置	

A12.6 影片与动画制作(Movies and Animation)

getframe	获得影片动画图像的帧	5 9 3
movie	播放影片动画	5 9 3
moviein	影片动画内存初始化	5 9 3

A12.7 其他图形操作指令(Miscellaneous)

ginput	从鼠标得到图形点坐标	5 8 2
graymon	设置缺省图形窗口为单色显示屏	
ishold	若图形处保持状态则为真	
rotate	在给定方向上旋转对象	
terminal	设置图形终端类型	
uiputfile	打开一个文件存储对话框	

uigetfile	打开一个文件查询对话框	
whitebg	设置图形窗口为白底	6 4 4
zoom	二维图形的变焦放大	5 8 1

A13 色彩控制和光照模式函数(Color Control and Lighting Model Functions)

A13.1 色彩控制(Color Controls)

caxis	色轴刻度	5 7 3;5 11 3
colormap	设置色图	5 7 1，5.7 2；5 11 3
shading	颜色渲染模式	5 3 1，5 7 4

A13.2 色图(Color Maps)

bone	蓝色调灰度图	5 7 2
cool	青和品红浓淡色图	5 7 2
copper	线性变化纯铜色调图	5 7 2
flag	红-白-蓝-黑交错色图	5 7 2
gray	线性灰度	5 7 2
hot	黑-红-黄-白交错色图	5 7 2
hsv	饱和色彩图	5 7 2
jet	变异 HSV 色图	5 7 2
pink	淡粉红色图	5 7 2
prism	光谱色图	5 7 2

A13.3 色图相关函数(Color Map Related Functions)

brighten	控制色彩的明暗	5 7 2
colorbar	色彩条状图	5 3 3，5 4
contrast	提高图象对比度的灰色图	
hsv2rgb	饱和色彩数据向红、绿、蓝数据转换	
rgbplot	取色曲线图	5 7 2
rgb2hsv	红、绿、蓝数据向饱和色彩数据转换	
spinmap	颜色周期性变化操纵	5 9 2

A13.4 光照模式(Lighting Models)

diffuse	漫反射表面系数	
specular	漫反射	
surfl	带光照的三维表面图	5 3 2，5 6 6
surfnorm	表面图的法线	

A14 特殊数学函数(Specialized Math Functions)

besseli	改进的第一类 Bessel 函数	3 5 1
besselj	第一类 Bessel 函数	
besselk	改进的第二类 Bessel 函数	
bessely	第二类 Bessel 函数	
betaBeta	函数	3 5 1
betainc	不完全的 Beta 函数	
betalnBeta	函数的对数	

cart2pol	直角坐标到极坐标的转换	
cart2sph	直角坐标到球坐标的转换 ellipj	
Jacobi	椭圆函数	351
ellipke	完全的椭圆函数	351
erf	误差函数	351；393
erfc	补误差函数	
erfcx	成比例的补误差函数	
erfinv	逆误差函数	351
expint	指数积分函数	
gamma	Gamma 函数	351
gammainc	不完全的 gamma 函数	
gammaln	Gamma 函数的对数	
gcd	最大公因子	
lcm	最小公倍数	
legendre	Legendre 伴随函数	
log2	基 2 剖分浮点数	
pol2cart	极坐标到直角坐标的转换	
pow2	基 2 标量浮点数	
rat	有理逼近	351；832
rats	有理输出	
sph2cart	球坐标到直角坐标的转换	

A15 特殊矩阵(Specialized Matrices)

compan	伴随矩阵	385;387
gallery	一些小测试矩阵	385
hadamard	Hadamard 矩阵	385
hankel	Hankel 矩阵	385
hilb	Hilbert 矩阵	385
invhilb	逆 Hilbert 矩阵	385
kron	张量积	385
magic	魔方阵	385
pascal	Pascal 矩阵	385
rosser	典型对称特征值实验问题	385
toeplitz	Toeplitz 矩阵	385
vander	Vandermonde 矩阵	385
wilkinson	Wilkinson's 对称特征值实验矩阵	385

A16 语言结构和调试指令(Language Constructs and Debugging)

A16.1 编程语言(MATLAB as a Programming Language)

eval	字符串宏指令	86
feval	函数宏指令	
function	函数文件头	822

global	定义全局变量	8 4 2
lasterr	最后一个错误信息	
script	MATLAB 命令文件	8 2 1
nargchk	输入变量个数检查	

A16.2 控制语句(Control Flow)

break	终止最内循环	8 5 5
else	同 IF 一起使用	8 3 2
elseif	同 IF 一起使用	8 3 2
end	结束 FOR、WHILE、IF 语句	
error	显示错误信息	
for	按规定次数重复执行语句	8 3 2
if	条件执行语句	8 3 2
return	返回	8.7 1
while	不确定次数重复执行语句	8.3 2

A16.3 交互式输入(Interactive Input)

input	提示键盘输入	8 5 2
keyboard	激活键盘做为命令文件	8 5 4
menu	创建菜单	
pause	暂停	8 5 3
uimenu	创建用户界面菜单	5 11 1 ; 5 11 2
uicontrol	创建用户界面控制	5 11 1 ; 5 11 2
yesinput	智能键盘输入	8 5 2

A16.4 调试指令(Debugging Commands)

dbclear	清除断点
dbcont	恢复执行
dbdown	改变局部工作内存的前后关系
dbquit	结束调试模式
dbstack	列出调用堆栈
dbstatus	列出所有的断点
dbstep	单步执行
dbstop	设置断点
dbtype	列出带行号的 M 文件
dbup	改变局部工作内存的前后关系
mexdebug	MEX-文件调试

A17 低层文件输入输出函数(Low-level File I/O Functions)

A17.1 文件的开启和关闭(File Opening and Closing)

fclose	关闭文件	
fopen	打开文件	8 8 1 ; 8 8 2

A17.2 无格式输入输出(Unformatted I/O)

fread	从文件中读二进制数据	8 8 1 , 8 8 2
fwrite	把二进制数据 写入文件	

A17.3 有格式输入输出(Formatted I/O)

fgetl	按行从文件读取数据并清除文件指针	
fgets	按行从文件读取数据并保留文件指针	
fprintf	把格式化数据 写入文件	
fscanf	从文件中读格式化数据	

A17.4 文件定位(File Positioning)

ferror	查询文件输入/输出错误状态	
feof	测试文件是否结束	
frewind	反绕文件	
fseek	设置文件指针	
ftell	得到文件指针	

A17.5 串的转换(String Conversion)

sprintf	写格式数据到串	
sscanf	在格式控制下读串	

A18 稀疏矩阵函数(Sparse Matrix Functions)

A18.1 基本稀疏矩阵(Elementary Sparse Matrices)

spdiags	从对角阵形成稀疏阵	3 12 2
speye	稀疏单位阵	
sprandn	稀疏随机阵	
sprandsym	稀疏对称随机阵	

A18.2 全阵与稀疏阵的转换(Full to Sparse Conversion)

find	寻找非零元素下标	
full	把稀疏阵转换成全元素阵	3 12 1
sparse	用非零元素和下标创建稀疏阵	3 12 1，3 12 2
spconvert	把稀疏阵转换到指定格式	3 12 2

A18.3 稀疏矩阵非零元的处理(Working with Nonzero Entries of Sparse Matrices)

issparse	如果是稀疏阵则为真	3 12 3
nnz	非零元素个数	3 12 3
nonzeros	非零元素	3 12.3
nzmax	为非零元素分配的存储器数	3 12 3
spalloc	为非零元素分配存储器	3 12 3
spfun	非零元素进行函数计算	3 12 3
spones	用 1 代替非零元素	3 12 3

A18.4 稀疏矩阵的可视化(Visualizing Sparse Matrices)

spy	稀疏矩阵的图形表示	3 12 3
gplot	以图论方式作图	

A18.5 排序算法(Reordering Algorithms)

colmmd	列最小度排序	3 12 3
colperm	基于非零算法列排序	3 12 3
dmperm	Dulmage-Mendelsohn 分解	3 12 3
randperm	随机置换向量	3 12 3
symmmd	对称最小度排序	3 12 3
symrcm	反向 Cuthill-McKee 排序	3 12 3

A18.6 范数、条件数和秩(Norm, Condition Number, and Rank)

condest	估计范1条件数	3 12 3
normest	估计2范数	3 12 3
sprank	结构秩	3 12 3

A18.7 树的操作(Operations on Trees)

etree	删除树
etreeplot	画出消除的树
treelayout	树的展开
treeplot	树的绘制

A18.8 关于稀疏矩阵的其他指令(Miscellaneous)

spaugment	形成最小二乘算法系统
spparms	对稀疏矩阵程序设置参数
symbfact	符号因子分析

A19 声音处理函数(Sound Processing Functions)

A19.1 常用声音函数(General Sound Functions)

saxis	声音轴刻度
sound	把向量转换成声音

A20 动态数据交换函数(DDE Client Functions)

ddeadv	设置咨询链接
ddeexec	送一个要执行的串
ddeinit	开始 DDE 对话
ddepoke	给应用程序发送数据
ddereq	从应用程序提取数据
ddeterm	终止
DDEddeunadv	释放咨询链接

A21 主启动文件(Local Function Library)

matlabrc	主起动文件
printopt	设置打印选择

A22 演示函数(Demonstration)

A22.1 引导(MATLAB/Introduction)

expo, demo	启动 MATLAB 演示屏幕	2 4 3
expomap	打开 MATLAB 演示主菜单图	

A22.2 矩阵(MATLAB/Matrices)

airfoil	据 NASA airfoil,显示稀疏阵	
buckydem	图以及其矩阵表示	
delsqdemo	不同域内的五点有限差分	
intro	MATLAB 入门	2 4 3
inverter	演示矩阵的逆	

matmanip	矩阵运算导论
sepdemo	有限元网格分选
sparsity	稀疏排序的演示

A22.3 数值计算(MATLAB/Numerics)

bench	MATLAB 基准测试程序
census	估计美国 2000 年的人口
eigmovie	求解特征值的动态演示
e2pi	e^pi 和 pi^e 哪个大?
fftdemo	快速离散傅里叶变换的应用
fitdemo	用单纯形算法的非线性拟合
fplotdemo	函数绘图
funfuns	演示功能函数运算
odedemo	常微分方程
quaddemo	求面积(定积分)
quake	地震波形
rrefmovie	化简行梯形形式的计算
spline2d	在平面样条曲线和图形点的获取演示
sunspots	答案是 11 08 的问题
zerodemo	用 fzero 函数寻找零点

A22.4 可视化(MATLAB/Visualization)

colormenu	选择色图
cplxdemo	复变函数图像
earthmap	地球的拓扑图
fourier	傅里叶级数展开
grafcplx	演示复数函数曲线
graf2d	演示二维曲线
graf2d2	演示三维曲线
imagedemo	图像演示
lorenz	Lorenz 吸引子
membran	创建 MathWorks's 标记
peaks	简单的双变量函数算例
penny	对硬币的不同观测
sqdemo	二次曲面的 UI 控制
vibes	L 型薄膜振动
xpklein	Klein 瓶
xpsound	演示声音特性

A22.5 图形对象底层操作语言(MATLAB/Language)

graf3d	演示表面图的柄图
hndlaxis	演示轴的柄图
hndlgraf	演示线图的柄图
xplang	MATLAB 语言的导论

A22.6 简单系统仿真(SIMULINK/Simple Systems)

bounce	弹跳球的 SIMULINK 系统演示

libintro	对 SIMULINK 模块库的入门介绍
onecart	一辆车的 SIMULINK 仿真演示
simintro	对 SIMULINK 的入门介绍
simppend	SIMULINK 系统单摆模拟
vdp	Van der Pol 方程的 SIMULINK 系统演示

A22.7 复杂系统仿真(SIMULINK/Complex Systems)

dblcart	两辆车的 SIMULINK 仿真演示
dblcart1	欠阻尼双车系统演示
dblpend1	演示双摆系统
1dblpend2	演示双摆系统 2
f14	F-14 的控制演示
penddemo	倒摆系统演示
thermo	温室控制系统的演示

A22.8 信号处理工具包(Toolbox/Signal Processing)

filtdem	信号滤波器演示
filtdem2	演示滤波器设计
sigdemo1	信号的离散傅里叶变换
sigdemo2	信号的连续傅里叶变换

A22.9 系统辨识工具包(Toolbox/System Identification)

iddems	系统辨识命令行的演示
sysiddm	辨识毛发干燥器的系统特性

A22.10 优化工具包(Toolbox/Optimization)

bandem	香蕉函数的优化
optdems	优化的命令行演示

A22.11 神经网络工具包(Toolbox/Neural Networks)

bckprp12	BP 网
bckprp62	带动量的 BP 网
neural	特征识别

A22.12 控制系统工具包(Toolbox/Control System)

ctrldems	控制系统命令行的演示
dskdemo	设置磁盘读/写头控制器

A22.13 鲁棒控制工具包(Toolbox/Robust Control)

accdm2	无摩擦弹簧连接小车系统演示
rctdems	鲁棒控制命令行的演示

A22.14 μ-分析和综合工具包(Toolbox/Mu-Analysis and Synthesis)

mudems	设置 μ-分析和综合命令行的演示
xpmu	描述 μ-分析和综合过程

A22.15 样条工具包(Toolbox/Spline)

spapidm2	样条插值
spldems	样条插值的命令行演示

A22.16 符号数学工具包(Toolbox/Symbolic Math)

xpcalc	微积分运算
xpgiv	Givens 变换

A22.17 图像处理工具包(Toolbox/Image Processing)

xpimage 图像处理性能的演示

A22.18 统计工具包(Toolbox/Statistics)

statdems 统计工具包的命令行演示

xppolytl 对有噪声的数据的交互式多项式拟合

A22.19 游戏(Extras/Games)

bblwrap 按钮猜测

life 生命游戏

xpbombs 扫雷游戏

A22.20 杂项(Extras/Miscellaneous)

crulspin 麻花圈旋转

logospin MathWorks'标记旋转动画

makevase 创建并画出镜像面

spinner 彩色线段空间旋转

travel 旅行推销员问题

truss 拱桥支架动画

wrldtrv 绕球飞行最大环

xpquad 超二次曲面

A22.21 与软件商的联络信息(Extras/Contact Info)

agents 国际区域联络信息的分布

contact1 怎样与 MathWorks 联络

contact2 怎样用电子信箱与 MathWorks 联络

contact3 怎样与 MathWorks 国际代理联络



参考文献

[1] MathWorks MATLAB Product Catalog 1996

[2] Matlnworks Matlab: The Ultimate Computing Environment 1996

[3] MathWorks MATLAB: News&Notes 1996

[4] MathWorks Matlab4 2c with Notebook 1995

[5] Pickover C A Exploring Infinite Realms IEEE Spectrum No.11 1995

[6] Braham R Math&Visulization (new tools new frontiers) IEEE Spectrum No 11 1995

[7] Braham R Application Sotfware IEEE Spectrum No 1:66－70 1995

[8] Biran A Breiner M MATLAB for Engineers 1995

[8] MathSoft Inc Mathcad 6 0 Plus 1995

[9] Wolfram Reasearch Inc Mathematica 2 2 1995

[10] Waterloo Maple Inc Maple V2 1994

[11] Riddle A Mathematical Power Tools IEEE Spectrum No 11:35－47 1994

[12] MathWorks Matlab User's Guide 1993

[13] MathWorks Matlab4 0 with Simulink1 2 1993

[14] Foster K R 'Abstract' Math Made Practical IEEE Spectrum No 11 1993

[15] Konmbluh K Seeing Data in Action IEEE Spectrum No 11 1993

[16] MathWorks Matlab3 5m 1992

[17] MathWorks Matlab3 0j 1988

[18] MathWorks Matlab User's Guide 1987



● 封面设计　许仲礼

● 封面制印　北京地大彩印厂




ISBN 7-81012-702-0


9 787810 127028 >

ISBN 7-81012-702-0/TP·243

定价：28.00元